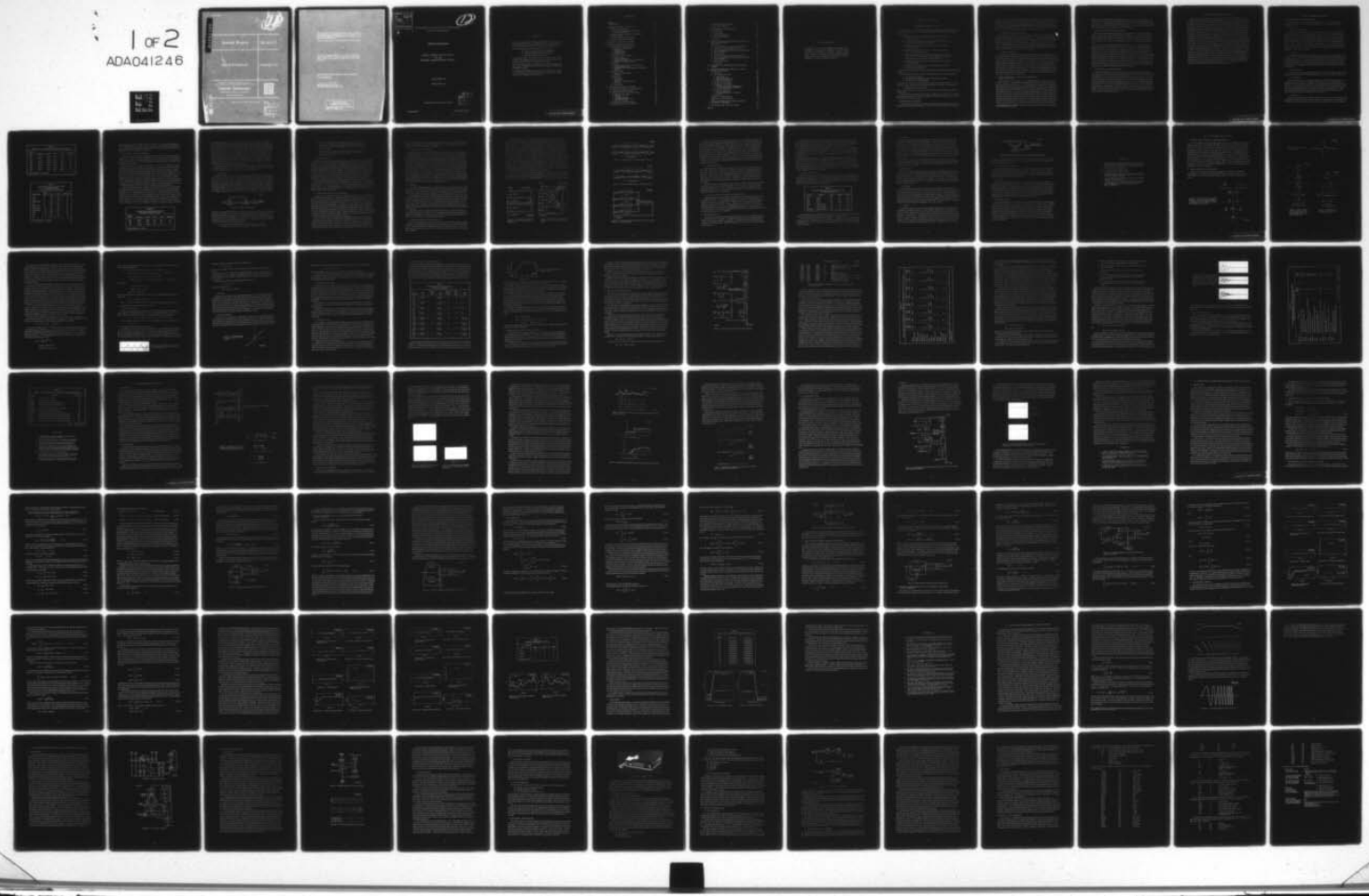
AD-A041 246 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 17/2
SPEECH EVALUATION.(U)
SEP 76 B GOLD
F19628-76-C-0002
UNCLASSIFIED ESD-TR-76-382 NL

1 OF 2
ADA041246



Annual Report

FY 1976-7T

Speech Evaluation

30 September 1976

Prepared for the Defense Communications Agency
under Electronic Systems Division Contract F19628-76-C-0002 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS

12

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

SPEECH EVALUATION

ANNUAL REPORT FOR FY 1976-7T
TO THE
DEFENSE COMMUNICATIONS AGENCY

30 SEPTEMBER 1976

ISSUED 16 MAY 1977

Approved for public release; distribution unlimited.

LEXINGTON MASSACHUSETTS

(See form 1473)

ABSTRACT

This volume reports the work performed during FY 76-7T on the DCA Speech Evaluation Contract. Work during this period on System Implications of Packetized Speech for DCA is reported under separate cover.

Three general areas of work are reported in this document:

(1) Work on narrowband terminal robustness;
(2) Work on wideband-narrowband tandeming; and
(3) Hardware speech-terminal efforts.

The robustness issues are defined early in this report; then, work on telephone-line simulation, robust pitch extraction, and operation of LPC vocoders in acoustically noisy environments is reported.

This report also discusses some approaches and progress made in the improvement of wideband devices, and the interoperability of wideband and narrowband terminals.

The design and development of a microprocessor-based LPC vocoder, as well as some work on the development of charge-transfer-device-based channel-vocoder equipment, also are described.

CONTENTS

ACKNOWLEDGMENTS

This report for FY 76-7T to the Defense Communications Agency was edited by J. Tierney and W. J. Finnegan. It is based on work by B. Gold, E. M. Hofstetter, R. J. McAulay, S. Seneff, J. Tierney, and O. C. Wheeler. The section on Charge Transfer Devices is based on work by R. W. Broderson of the Electronics Research Laboratory, University of California, Berkeley.

SPEECH EVALUATION

I. PROGRAM OVERVIEW – FY 1976-7T

Lincoln Laboratory's commitment to the Defense Communications Agency during the period FY 76-7T was outlined in the FY 75 final report and was incorporated into the DCA statement of work as follows:

(a) Tandeming and conferencing experiments using high-, medium-, and low-rate digitizers.

(b) Telephone channel simulation for controlled testing of phone-line distortions and their effect on speech digitizers.

(c) Investigations of speech digitizer talker sensitivity, and techniques for reducing this effect.

(d) Study of medium-rate coders with a view toward improved quality and/or lower implementation costs.

(e) Design and development of a low-cost LPC microprocessor terminal for use at 4.8, 3.6, and 2.4 kbps. Such a terminal must be suitable for large-scale defense communication systems deployment.

(f) Investigation of effects and cures for carbon button microphone inputs to speech digitizers.

(g) Investigation of effects and cures for input environmental noise vulnerability in speech digitizers.

All these tasks are directly or indirectly related to DCA and DoD Secure Speech Consortium interests.

For reporting purposes, these tasks are grouped into three major technical categories:

A. Robust Speech Processing
B. Interoperability of Wideband and Narrowband Speech Terminals
C. Vocoder Hardware Implementations.

In this Overview, we discuss the rationale for the selection of these three topics as the major research areas and summarize the conclusions derived from our findings. A fourth topic:

D. Experiments and Demonstrations

includes certain field activities at DCEC and NRL to demonstrate accomplishments under the contract tasks.

In Sec. II, we list more specifically the tasks accomplished; the remainder of the report gives detailed information on these tasks, leaning heavily on reports prepared during FY 76-7T under the Speech Evaluation contract.

A. ROBUST SPEECH PROCESSING

It is now generally accepted that a variety of narrowband speech digitizers work quite satisfactorily under laboratory environments. In this effort, we were concerned with improving

performance under degraded environments such as acoustically coupled background noise, telephone system channel errors, input distortions of carbon button microphones, and problems of atypical talker characteristics. The Consortium Test and Evaluation (T&E) program results had already convincingly demonstrated that intelligibility was often significantly reduced due to these degradations. During the course of FY 76-7T, we were able to work on the problems of acoustically coupled background noise and telephone input speech.

1. Acoustically Coupled Background Noise

The major activity in this area was directed toward modeling the noise background as a basis for applying the methods of statistical decision theory. Attention was specifically directed toward the voiced-unvoiced decision problem since, in a noisy environment, this component of speech analysis was judged most likely to degrade. Furthermore, experience has shown that failure of this component causes a drastic reduction in listener acceptability of the vocoder system. A statistical decision criterion has been successfully applied (in non-real-time) in the presence of correlated Gaussian noise background, and a conceptual design toward a real-time implementation of the new algorithm has been outlined.

2. Telephone Input Speech

It has long been recognized that speech distortions introduced by the telephone handset and by a variety of telephone channel phenomena can adversely affect the performance of a narrowband speech processor. One of the major difficulties in approaching this problem has been the fact that telephone channels differ greatly. Our first step was thus to set up a real-time flexible telephone channel simulation on a Digital Voice Terminal (DVT). This facility allows for controlled variations of telephone distortions caused by frequency response, phase distortions, frequency offsets, and nonlinear effects. Our next step was to investigate the pitch measurement problem of a vocoder for telephone speech.* This led to an innovative design of a pitch detector based on a harmonic analysis of the speech wave; our work showed that this form of preprocessing resulted in a pitch algorithm that was significantly less vulnerable to both telephone phase distortion and background noise.

B. INTEROPERABILITY OF WIDEBAND AND NARROWBAND SPEECH TERMINALS

The Autosevocom II network will consist of CVSD terminals interconnected by 16-kbps channels. However, it is desirable that a significant number of users with lower available bandwidth be able to use the network. For a narrowband terminal to speak to a wideband terminal requires appropriate interfacing at the switch between two distinct speech algorithms. One specific method of providing this interface is via tandem connections; this is the method we investigated in FY 76-7T. Since LPC is presently considered to be the most suitable narrowband speech digitizer, a group of experiments involving CVSD-LPC tandems were implemented. The overall results indicated that such a tandem, in either direction, could be made acceptable provided that LPC was run at rates of 3600 bps or higher, the CVSD parameters were carefully adjusted, the CVSD channel error rate was less than 1 percent, and the voice volume was kept reasonably steady. Thus, our overall conclusions were that LPC-CVSD tandeming could be acceptable under a limited set of benign conditions, but that tandeming was

* This undertaking was supported in part by the Defense Advanced Research Projects Agency of the United States Government.

significantly less robust than either component of the tandem. Two specific "remedies" were tried: one was to insert a chirp filter between the LPC and CVSD in a configuration where the LPC comes first – modest quality improvements were noted; the other was to replace the CVSD algorithm with a 16-kbps APC algorithm. In our judgment, the tandem of LPC and 16-kbps APC resulted in significant improvement over the LPC-CVSD tandem.

C. VOCODER HARDWARE IMPLEMENTATIONS

The recent introduction into the marketplace of high-speed LSI chips made it feasible to design a small microprocessor dedicated to the LPC algorithm. This task was accomplished and is documented later in this report; at present, two working models exist and one has been used for demonstration at DCEC. We believe that, at the time of completion of this project, it represented the most compact and potentially least costly narrowband speech processor yet built.

As a result of the favorable performance characteristics of the U.K. Belgard channel vocoder noted in the Narrowband Speech Consortium T&E program, a Belgard vocoder was implemented on the DVT. Comparison results indicated Belgard to be competitive with LPC in terms of voice quality and robustness. Belgard implementation on digital machines requires more than twice the signal-processing power of LPC implementation. However, new technology such as charge transfer devices and efficient MOS integration makes special-purpose hardware realizations of transversal filters and detectors attractive. In this light, a 3-month effort was subcontracted to the University of California, Berkeley, Electronics Research Laboratory to implement a full-wave rectifier and desampling filter for channel vocoder use. This circuit has been delivered to Lincoln for evaluation. In addition, a study at both Lincoln and Berkeley of a full vocoder channel analyzer, consisting of a bandpass filter, a rectifier, and a low-pass filter, indicates an efficient realization to be feasible.

D. EXPERIMENTS AND DEMONSTRATIONS

Several demonstrations and tests were set up and run at both DCEC-Reston and NRL-Washington. Narrowband consortium tests continued until early September 1975 when the DCA DVT equipments were moved to NRL for HF channel tests of the LPC vocoders. At the end of September 1975, the DVTs were returned to Reston for vocoder demonstrations, and were returned in February 1976 to Lincoln Laboratory. In June 1976, two DVTs equipped with read-only memory (ROM) to make them freestanding vocoders at 2.4, 3.6, and 4.8 kbps (as opposed to being loaded from a PDP 11/20 as in earlier tests) were delivered to DCEC for additional demonstrations. These units were returned to Lincoln in midsummer. At the end of FY 7T a tandem LPC-chirp filter-CVSD experiment consisting of a DVT with ROM implementing a chirp filter, a CVSD encoder decoder, an error generator, and the newly developed microprocessor LPCM vocoder were delivered to Reston.

II. OUTLINE OF THIS ANNUAL REPORT

The sections which follow describe each of the major areas of effort for FY 76-7T, plus our conclusions and a statement on our continuing efforts. The area of "robustness" of narrowband speech digitizers is discussed in Sec. III. Starting from the Narrowband Speech Consortium T&E results, the remaining serious problems of vocoder sensitivity are outlined. In Sec. IV, the results of a telephone-line simulator programmed on the Lincoln Laboratory DVT signal-processing machine are presented. The simulation runs in real time, allowing an experimenter to add standard phone-line distortions to a speech signal under study before processing the signal with a narrowband device. Section V shows an application of the telephone-line simulator. In particular, this Section discusses development of a fundamental frequency/pitch extractor for telephone-line distorted speech. A device of this type is a necessary part of present narrowband speech terminals. An approach to the problem of LPC vocoding of noisy speech is presented in Sec. VI. The noisy signal may have been produced by passage through a noisy channel or by a talker in a bad acoustic environment. This work is complementary to the pitch detection work of Sec. V, since both aim toward vocoding of noisy, distorted speech signals. Section VII attempts to summarize the various investigations toward improved CVSD, improved tandeming of CVSD, improved tandeming of LPC-CVSD, and an improved high-rate (16-kbps) APC system for higher quality narrowband-wideband tandeming. This Section also presents some results on dispersive, nonrecursive, digital, all-pass, chirp filters for speech conditioning between narrowband and wideband terminals. Section VIII reports the results of the LPCM hardware development; this program has produced two microprocessor-based LPC speech terminals. The design and development process as well as the final specifications of the existing devices are also discussed here. Section IX is concerned with our small outside effort in charge transfer device-MOS implementation of a Belgard-like channel vocoder. A statement of future work based on the accomplishments described here is presented in Sec. X.

IV. THE TELEPHONE-LINE SIMULATOR

III. ROBUST NARROWBAND DIGITIZERS

A. IDENTIFICATION OF ROBUSTNESS ISSUES

Partly through the Test and Evaluation (T&E) program of the Narrowband Speech Consortium, and partly through work in our laboratory, we can identify the following "robustness" issues.

1. Variation Among Speakers

We observe, in demonstrating our various speech algorithms to visitors, that there is a wide variation in what listeners find acceptable; this variation seems to depend on both the speaker and the listener. For example, we hear comments like "I read you but it doesn't sound like you," whereas another listener will be very happy with the same speaker's identity. Many of the T&E tests on our LPC system show a definite drop in intelligibility in going from male to female speakers. If a speaker speaks too softly (or too loudly), degradation usually results. It is a commonly accepted folk-fact that people who have been in the narrowband voice business for awhile make better speakers. To summarize, it is quite well accepted that narrowband systems such as LPC are less robust than a telephone line and may even be unacceptable for a large number of speakers.

2. Tandeming

Communications networks tend to grow partly in a random manner, and it is expected that any large-scale network will include channels of differing bandwidths and will therefore require different speech terminals. This fact, plus the requirement of conferencing, leads to the tandeming problem wherein, for example, the voice must be processed through an LPC followed by a delta-modulation system. T&E results have demonstrated that tandeming of two speech processors degrades intelligibility and quality, and that three or more in tandem are generally unacceptable.

3. Background Noise

In many military environments, high-level background noise is unavoidable. Again, severe degradation often leading to unacceptable results has been demonstrated from the T&E results.

4. Telephone Speech

For the foreseeable future, we expect the telephone handset to be the most popular and cheapest of all speech terminals. Flexibility of both military and civil telephony would be greatly enhanced if existing analog facilities could be integrated with new digital communications facilities. To accomplish this requires that speech algorithms remain robust after the analog voice has been processed through a carbon button microphone and a telephone line. To date there is insufficient quantitative data on the effects of such processing, although it appears from informal experiments that degradation can be very severe.

5. Channel Errors

Channel errors which perturb the speech processor digital output can cause varying trouble depending on the robustness of the system to such errors. One approach to such problems is to focus attention on the modem and try to reduce the channel error rate to a very low number by

TABLE III-1

COMPARISON OF MALE-FEMALE DRT SCORES FOR LINCOLN LPC

Data Rate (bps)	Microphone	Noise	Male	Female	Difference*
2400	Dynamic	Quiet	83.7	77.7	6.0
2400	Carbon	Quiet	79.4	74.7	4.7
2400	Dynamic	Office	82.6	76.0	6.6
3600	Dynamic	Quiet	86.1	80.4	5.7
3600	Carbon	Quiet	81.6	74.9	6.7
4800	Dynamic	Quiet	88.2	82.7	5.5
4800	Carbon	Quiet	85.5	82.4	3.1

*Average difference = 5.47 percent.

TABLE III-2

COMPARISON OF MALE-FEMALE DRT SCORES FOR HIGH-RATE SYSTEMS
(Dynamic microphones, quiet background)

System Name	Rate (kbps)	Male	Female	Difference*
CVSD	32	95.7	93.6	2.1
CVSD	16	91.0	88.0	2.2
ARC (CODEX)	16	90.8	87.8	3.0
HY11	16	90.2	86.7	3.5
ARC (Lincoln)	16	92.2	91.5	0.7
ARC (Lincoln)	9.6	87.3	87.9	−0.6
ARC (CODEX)	9.6	85.3	84.6	0.7
CVSD	9.6	82.3	75.7	6.6
HY11	9.6	79.7	75.7	4.0
APC	8	89.3	87.6	1.7

*Average difference = 2.39 percent.

means of redundant coding techniques. However, economies may be gained by designing the source coding to be robust in a speech sense. In particular, demanding error-free transmission in a jamming environment may be more costly than designing a speech processor which degrades gracefully as channel errors increase.

B. THE SPEAKER VARIATION PROBLEM

At present, there does not seem to be enough T&E data to allow us to establish quantitative reasons for the accepted fact that LPC systems are, in general, not robust across speakers. However, even a cursory examination shows that the T&E female speakers scored consistently lower than the males, so let us direct our attention to the possible reasons for this special example of speaker variation.

Table III-1 shows some available T&E results for the Lincoln LPC algorithm as implemented on the LDVT (Lincoln Digital Voice Terminal). This table shows that for both carbon and dynamic microphones, for different data rates and even in one given noisy environment, the Lincoln LPC discriminates quite consistently against women for DRT (diagnostic rhyme test). Let us inquire into some possible reasons. The whole problem can be speculated out of existence by assuming that the particular females chosen for T&E speakers simply did not articulate as clearly as the males, and thus the Lincoln LPC was not to blame. A look at Table III-2 shows that the females scored somewhat lower for all but one of the tested wideband (9.6 to 32 kbps) systems. However, the average drop for Table III-2 was 2.39 percent, which is little more than the standard error, whereas for Table III-1 the average drop was 5.47 percent. The difference between Tables III-1 and III-2 is certainly statistically significant!

An obvious potentially significant parameter which might help explain this is the bandwidth. For example, we know that, for the voiceless fricative sounds (s, sh, f, th), female spectra generally extend to a higher bandwidth than male spectra. Increasing processing bandwidth increases the cost of the processor in a very direct way by forcing an extension of its computational power; thus, proposals for increased bandwidth should be viewed very cautiously. There is some slight evidence that increased bandwidth helps females from the scanty Belgard results shown in Table III-3. This 2400-bps channel vocoder utilizes substantially more bandwidth than the Lincoln LPC, both in its voicing detector system as well as in the spectral processing. For the two quiet cases shown, the scoring loss for females is significantly lower. The large

TABLE III-3

COMPARISON OF MALE-FEMALE DRT SCORES FOR BELGARD CHANNEL VOCODER

Data Rate (bps)	Microphone	Noise	Male	Female	Difference*
2400	Dynamic	Quiet	87.2	84.6	2.6
2400	Carbon	Quiet	82.6	80.2	2.4
2400	Dynamic	Office	84.6	77.6	7.0

*Average difference = 4 percent.

scoring loss for females for office noise could be attributed to a clobbering of those high frequencies which we speculated are needed to raise female scores. Apparently contradictory results are obtained from a perusal of Table III-2, where female scoring was not significantly lower, even though most of these higher data-rate systems processed equal or lower bandwidth speech than the Lincoln LPC. The most plausible theory which is consistent with Tables III-1, III-2, and III-3 is this: for a high data-rate system, where the speaker's articulation is reproduced quite accurately, enough transitional acoustic cues remain in the processed speech so that the average listener can score well even in the absence of the higher frequency components. However, for a lower data-rate system these cues (such as formant transitions) are sufficiently lacking so that a higher score can be brought in "by the back door" by simply incorporating higher frequencies. Thus, we can hope that algorithmic improvements in LPC will avoid the need to process more speech bandwidth.

If bandwidth is not the main issue, other pre-computer audio processing functions may yet be of first-class importance. Two items come to mind immediately, namely, frequency equalization (pre-emphasis) and volume control. Let us discuss these items.

1. Volume Control

We know that speech sounds vary in volume by about 40 dB, from the powerful vowel a as in "fat" to the weak fricative f. Both analog and digital speech processors can cope with this dynamic range, but if we superpose variations in the average volume from very quiet speakers to those who roar into the handset, this gives us another 20 to 30 dB to deal with. From our experience, we have grave reservations about the use of commercially available volume-control circuits for audio pre-processing. Such devices cause distortions and make the problem of isolating and dealing with many distortions caused by the narrowband processor more difficult. Also from our experience, we feel quite confident that monitoring of the input speech volume by a competent human monitor could go a long way toward avoiding the overflow and underflow

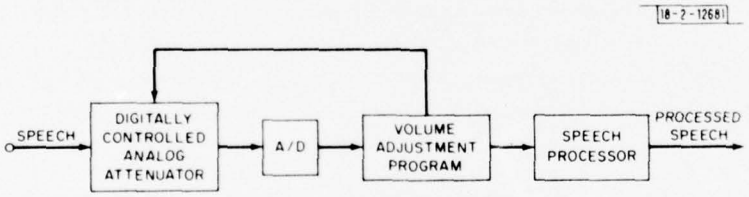


Fig. III-1. "Small" volume control.

problems in the digital speech processor. Figure III-1 indicates a setup which at least permits research on this problem. In this figure, we are ignoring other audio conditioning such as a pre-emphasis or pre-sampling filter. By controlling a commercially available digitally controlled analog attenuator via a program in the processor, much more flexible (i.e., "smarter") volume control can be incorporated into the processor. For example, the program would not have difficulty implementing the following concepts:

(a) Every time an input speech sample is peak clipped on its way into the A-D converter, the attenuator is switched to 6 dB (1 bit) lower volume.

(b) If several seconds of processing indicate that no speech was entered into the system during that time, the attenuator can remain untouched on the basis that no speech information was available to make volume-control decisions.

(c) If several seconds of speech show that the upper bits of the A-D converter have not been tickled at all, the attenuator can be switched to permit greater volume.

2. Spectrum Equalization

This has been the subject of continuing controversy. Presently, there is not even complete agreement as to whether pre-emphasis helps or hinders LPC processing, although the predominant bias favors the use of pre-emphasis. Adaptive pre-emphasis poses a problem since, at the synthesizer, compensating adaptive de-emphasis must be incorporated based on the transmitted pre-emphasis parameter (or parameters). Notice that such extra transmission is not needed for volume control. It is interesting that several experienced narrowband speech workers favor the incorporation of fixed pre-emphasis and no de-emphasis. This fits in with the "back-door" theory discussed above wherein poor processing is compensated by letting in excessive high frequencies. If not for possible adverse effects in the presence of background noise plus utter degradation in tandem situations, this might not be too bad an answer.

The biggest objection against adaptive pre-emphasis is the difficulty of suitably measuring the desired adaptive parameter, and the subsequent possibility of doing more harm than good. A simple step to try would be to incorporate two fixed pre-emphasis filters, one for males and one for females. Then, analysis would consist of (a) making a male-female decision, (b) transmitting a single bit, and (c) switching between one of two possible pre- and de-emphasis networks at the transmitter and receiver.

3. Adaptive Quantization of the LPC Parameters

At present, the LPC parameters chosen for transmission are the pitch, voicing decision, gain, and reflection coefficients. To date, there has been no convincing argument for or against the use of these parameters except the practical one that no better set has been found. It is an interesting fact that channel vocoder parameters consist of the sampled spectral magnitude or some linear function of these samples. It also seems to be true, at the moment, that the best 2400-bps channel vocoder outperforms all 2400-bps LPC implementations in the intelligibility tests of the T&E program. Over the years, important intuitions have been built up about the way speech spectral magnitude changes with both frequency and time. In either dimension there are fairly well-understood constraints on these changes, and thus clever adaptive coding schemes can be and have been successfully developed. As yet, no correspondingly sophisticated coding scheme has been demonstrated for reflection coefficients.

There is a sound reason why the apparently straightforward problem of quantizing LPC (or channel vocoder) parameters seems to be the most difficult and time-consuming problem in the design of narrowband speech terminals. Certainly, for 2400-bps systems, the greatest injustice to the spectral integrity of the speech occurs at this very quantizer. Makhoul[1] has shown that spectral sensitivity to reflection coefficient (k_i) changes is greatest when $|k_i| \approx 1$; unfortunately, histograms of k_i show clearly that $|k_i|$ is rarely in the vicinity of unity. Also,

Seneff[2] has demonstrated that a coding scheme which utilizes a given speaker's k_i histograms improves the LPC output. The two above techniques are actually contradictory, and no one has yet offered a way of resolving this contradiction.

4. Philosophy of Speaker Adaptation

Although no clear-cut solutions arise from the above discussion, a conceptual approach to the problem does emerge. It is clear that significant variations in the voice to be processed by a narrowband algorithm occur due to the innate differences among speakers, different microphones, differing ways whereby speakers handle the handset, different pre-emphasis, post-emphasis, pre-sampling, post-sampling filtering, different sampling rates, etc. It is very unlikely that all these variations can ever be standardized. What is needed is an adaption mechanism. Philosophically, at least, this can be introduced by defining a third basic epoch (in addition to a sampling epoch and a frame epoch) which we could call an adaptation epoch. Intuitively, this epoch should be of about 1-sec duration. The system would be collecting statistical data during each such epoch and adjusting system parameters (as opposed to voice analysis parameters) from epoch to epoch. Obviously, this type of adaptivity is useful only for conversational speech communication, and would perhaps be a hindrance for items such as word list intelligibility testing (unless one insisted on extended testing with a single speaker). From an implementation point of view, one would guess that a fairly extensive amount of additional processing would be needed, but at a slow rate. The difficult part of the work would be the invention of algorithms that would do more good than harm.

C. TANDEMING

Since a single narrowband digital voice terminal degrades the clear input speech, it makes sense that two or more such devices in tandem will cause more degradation. The T&E results bear this out. For example, the Lincoln LPC DRT score goes from 83.7 through one system to 72.6 for two, and to 66.5 for three in tandem. The PAR scores go from 55.4 to 45.0 to 13.7.

A surprising result emerged by comparing a tandem connection of two Lincoln LPCs with an LPC-CVSD tandem. Since CVSD is a wideband (16-kbps) system, the latter tandem would be expected to be an improvement over the tandeming of two narrowband systems. The actual results were opposite, however, with LPC-CVSD yielding a DRT of 70.0 compared with 72.6 for LPC-LPC. Also, LPC-LPC-LPC gave a DRT score of 66.5 compared with 61.7 for the worst combination of a single LPC with two CVSDs. The clear indication is that the CVSD algorithm (at least with the presently used parameters) can't cope too well with the reproduction of LPC speech.

A wide variety of tandem configurations exists which could be of practical interest in a communications network. At present, we are not equipped with a general methodology for improving the voice quality of an arbitrary tandem of the same or different speech algorithms. However, we can address several specific pertinent topics.

1. Conferencing of Digitally Processed Speech

Ordinary analog telephonic conferencing is attained through summing the individual speech signals at a central node and distributing the sum to all the conferees. From the speech point of view, conference calls lead to a degradation in signal-to-noise ratio (SNR). Most of the time, only one conferee is speaking, yet the environmental noise associated with each conferee is always present.

It is interesting to inquire as to whether selection algorithms which tend to favor the "louder" speaker would benefit the above analog conferencing. A device which selected the "loudest" speaker and suppressed the signals on the other lines would certainly improve the received SNRs, but could conceivably cause other problems resulting from abrupt switching among lines. Since we are not familiar with any comparative work that may have been done between these two conferencing methods, we want to make only the following point: for analog conferencing, summing algorithms are simpler to implement than selection algorithms and, except for signal-to-noise loss and telephone-line degradation, no other processing loss in speech quality takes place. The situation is quite different, however, for conferencing which involves speech digitizers such as CVSD or LPC. First, use of the summing algorithm implies that tandeming is necessary. Second, digital selection hardware appears to be cheaper than the hardware needed to implement analog summing of digital speech. Figure III-2 shows an example of the use of the summing algorithm for four digitally processed speech signals. In order to sum the speech of the conferees, each input bit stream from an analyzer must first be synthesized; then the four resulting signals must be added, and the sum analyzed and distributed to all conferee synthesizers. Thus, a quantity of hardware is required at the conferencing node. Also, each conferee's speech is processed through two algorithms.

In Fig. III-3 the synthesizers of Fig. III-2 are replaced by a selection device which chooses one of the conferees to send on. The selection algorithm and the switch should, in general, be appreciably simpler than the collection of synthesizers required for Fig. III-2. Also, each speech signal travels through only a single processor so that tandeming degradation is removed.

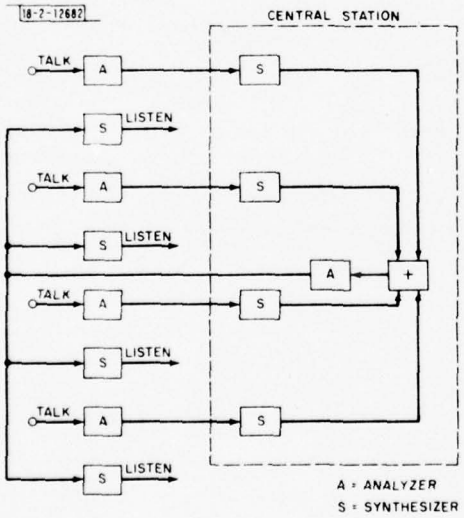


Fig. III-2. Summing algorithm for four digitally processed speech signals in a conference.

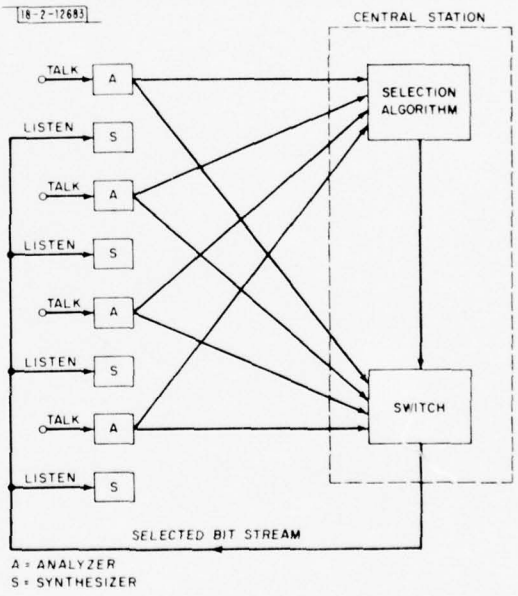


Fig. III-3. Digital conferencing using an algorithm to route a single selected bit stream.

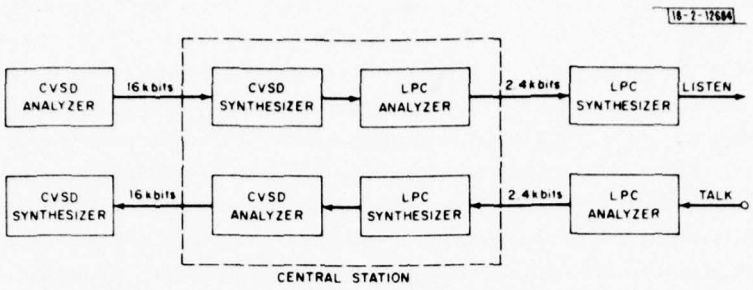


Fig. III-4. Configuration for a "disadvantaged" LPC user.

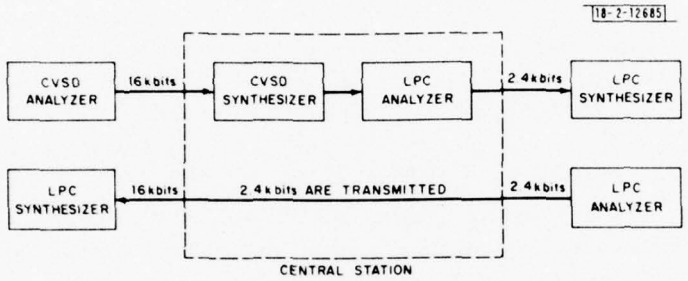


Fig. III-5. Alternate configuration for a disadvantaged user.

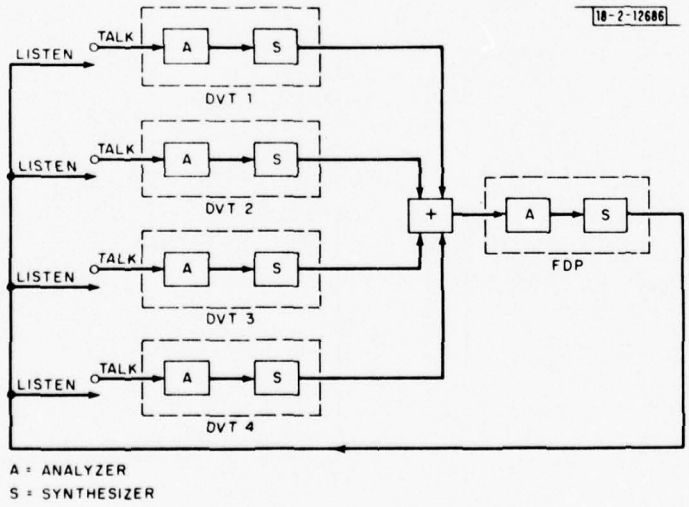


Fig. III-6. Trick for real-time simulation of four-way conference using four DVTs and the FDP.

A conferencing strategy which involves an overhead cost of accompanying control information but leads to simpler speech-processing hardware has been quite successfully demonstrated in a recent experiment designed by J. Forgie.[3] Each conferee has a control box with a red and green button plus a red and green light. Pressing the green button signifies a desire to talk, while pressing the red button signifies that the speaker is about to stop. The green light permits the conferee to talk (and be heard), while the red light indicates that he should not talk (or that if he does, no one will hear him). Management of the resulting control information and channel allocation is handled by computer. This setup leads to "polite" conferencing, with no opportunity for interruptions. Intuitively, we feel that the ability to interrupt becomes less desirable as the network delay increases, as it would through a satellite link or the ARPANET. Even delays caused by long-distance lines plus associated switches could be sufficient to make Forgie's method attractive.

2. The Disadvantaged User

In many long-distance situations, both users in a point-to-point voice communication do not have the same bandwidth. Consider a user who must cope with a 2400-bps channel and thus has an LPC terminal. On the other end is a "wealthy" user with a 16-kbit channel at his disposal and a CVSD terminal. The straightforward configuration for this situation is shown in Fig. III-4. It is clear that for each CVSD-LPC conversation, a complete duplication of the terminal equipment is needed at the central node. Furthermore, this configuration results in both CVSD-LPC and LPC-CVSD tandeming, with the latter resulting in poor scores from the T&E results.

A partial alleviation of this situation, at some extra cost, is indicated in Fig. III-5 – namely, to provide the CVSD user with an LPC synthesizer. Then the CVSD → LPC tandem remains, but the worst offender, LPC → CVSD, is now gone. As yet, we have no information on the potential production cost of a synthesizer, but we know that two of the costlier items – the correlator and pitch detector – are not present.

3. Remarks on Real-Time Simulation of Tandeming Situations

Conferencing protocols and strategies should evolve based on experiments; in this section, we confine our remarks to the capabilities at Lincoln for simulating the situation shown in Figs. III-1 through III-5. A configuration involving four DVTs and the FDP has the performance capability needed to simulate all but Fig. III-2. However, we can manage even this case by means of the trick shown in Fig. III-6. Comparing Figs. III-2 and III-6 we see that the effects of tandeming LPCs plus the effort of several simultaneous inputs into an LPC analyzer are correctly simulated. Thus, four DVTs plus the FDP can handle all situations we have described up to and including a 4-way LPC conference.

D. BACKGROUND NOISE

We should first realize that different kinds of background noise create different problems, and must be dealt with accordingly. The background noise used in the T&E program included "office" noise (background speech, typewriters, etc.), airborne command post (ABCP) noise, ship noise, and helicopter noise. Before discussing the T&E results, it is worth making a few general remarks.

"Office" noise is not a major problem, in our opinion, since these noise sources are usually of no greater volume than that of the talker and, being much further from the microphone, do not greatly alter the input SNR. For example, the Lincoln LPC suffered little or no loss in intelligibility and quality in going from a quiet to an office environment.

Helicopter noise causes disastrous intelligibility and quality losses in the tested speech. This noise interferes greatly with the excitation analysis, but also may cause large problems in extracting the vocal-tract LPC parameters. In our opinion, this problem should be treated in a special way dictated by the environment, and any fixes should not be expected to work for other noisy cases.

ABCP and ship noises appear to be more straightforward types of noise which can be modeled as colored Gaussian noise. The most crucial aspect, we think, is inclusion of the special characteristic of the noise-canceling microphones. We are quite convinced from our own experiences that failure to match the pre-emphasis filter and/or the quantization levels of the LPC parameters to the microphone causes large differences in the result. It is known that noise-canceling microphones have response characteristics that are very different from either carbon or close-talking dynamic microphones.

1. Summary of T&E Results

Table III-4 summarizes the presently available T&E results of the background noise testing of the Lincoln LPC system. For the 2400- and 3600-bps cases, both ship and ABCP cases drop below the acceptable DRT score. For 4800 bps, intelligibility holds up well enough to be acceptable, which lends credence to our argument that quantizer levels for LPC parameters should be adjusted according to the microphone used.

TABLE III-4
DRT SCORES FOR LINCOLN LPC FOR VARIOUS TYPES OF NOISE BACKGROUND

Environment	Microphone	2400 bps	3600 bps	4800 bps
Quiet	Dynamic	83.7	86.1	88.2
Quiet	Carbon	79.4	81.6	85.5
Office	Dynamic	82.6	86.0	85.8
ABCP	Noise canceling	72.6	73.7	79.3
Ship	Noise canceling	70.1	73.1	81.3
Helicopter	Noise canceling	48.0	45.0	56.0

The table shows clearly that helicopter noise is in a class by itself, and causes such severe speech-processing problems that it ought to be treated as a separate problem.

For all the noise backgrounds that caused significant degradation, noise canceling microphones were used. Unfortunately, we have no results using these same microphones in a quiet environment so that it is not clear to what extent the lower scores were caused by the noise or by the microphone.

2. Comments

It is difficult to propose solutions to problems involving signals in noise without having a good deal of knowledge about the nature of the noise. Using a Gaussian colored noise model which is additive to the signal, McAulay[4] proposed how the maximum-likelihood method implicit in quiet LPC can be extended to that of finding LPC parameters in a noise background.

Noise makes it very difficult to design a good voiced-unvoiced decision algorithm. Aside from the poor resultant speech quality, background noise makes "silence" detection very difficult – by silence we mean the absence of speech. Accurate detection of silence makes possible TASI-like systems with consequent savings of as much as 2:1 or 3:1 in communication bandwidth! Thus, a fruitful research area is that of applying maximum-likelihood methods of detection of speech vs noise.

Summarizing, it seems that background noise in LPC can be combated by a variety of methods, including proper choice of microphone, audio pre-processing, adaptive-quantization, and maximum-likelihood techniques. Accumulation and categorization of a noise data base is an important aspect of such work.

E. CHANNEL NOISE

There are two important aspects to the investigation of channel noise effects on speech terminals. For one set of situations, sophisticated modem technology allows effectively error-less transmission until the noise exceeds a fairly sharp threshold. In this case, the speech terminal will either work or quit working, and the issue of the sensitivity of the particular algorithm to channel noise loses importance. In another set of situations, it may be impractical to include sophisticated modems, and then vulnerability to channel errors can be an important feature of a speech processor.

1. T&E Results

The T&E results give some data on the lowering of intelligibility and quality scores for channel error rates of 1 and 5 percent. A general statement can be made that a 1-percent error rate causes little perturbation of the LPC scores, whereas 5 percent causes a significant and most likely unacceptable degradation in both intelligibility and quality. Somewhat mysteriously, the performance of the 2400-bps Belgard channel vocoder suffered little loss of intelligibility at 5-percent error rate; at the moment, we have no good explanation and would like to further investigate possible differences in vulnerability between LPC and channel vocoders. As expected, CVSD systems also suffered little degradation at these error rates.

2. Jamming

From a practical point of view, the most interesting problem involving channel errors is the problem of combating a jammer in a line-of-sight or satellite link. In our opinion, this situation should be explored in the context of both an adaptive modem and an adaptive speech processor. A mechanism for maintaining voice communications over a jammed link as long as possible is shown in Fig. III-7. In addition to the spread spectrum modem and the speech processor, a technique is needed for probing the channel and estimating the instantaneous channel capacity; a set of algorithms is presently being developed by Goodman et al.[5] to do just this. The set of speech algorithms already developed on the DVT encompasses data rates from 2.4

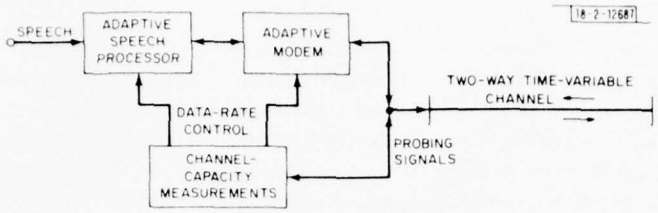


Fig. III-7. Method of probing a channel to estimate channel capacity and dynamically adapting the speech processor to the available rate.

to 64 kbps; thus, a variable data-rate source-coding system is presently a reality. It remains to determine appropriate variable rate spread spectrum techniques and combine all elements into a real-time simulation system.

3. Selective Coding of Parameters

From a perception viewpoint, we expect that some systems parameters cause worse degradation if in error than other parameters; for example, a pitch error in LPC may be more harmful than a comparable error in an LPC parameter. Flanagan[6] has performed just perceptible difference measurements on formant errors, and perhaps a similar study would be in order on the effects of LPC parameter errors.

4. Summary

Rather than try to make speech coders less vulnerable to channel noise, it seems more useful to reduce channel errors by redundancy coding techniques. Both approaches presumably require extra hardware, but coding techniques are a fully developed field so that no new algorithms need be developed. Experiments on selective coding of parameters are still worth performing; a good start would be comparison of pitch vs spectral sensitivity in either vocoders or LPC systems. Finally, the proven fact that the DVT is a flexible and easily adaptive speech processor plus the ongoing variable channel capacity work make it very tempting to develop a simulation facility for a speech terminal which adapts well to jamming.

F. TELEPHONE SPEECH

It has been known for many years that speech that has been transmitted through a telephone channel prior to vocoding causes very significant degradation of the vocoded speech. Admittedly this is a vague statement, since telephone channels have great variability. Trying to find a common formulation for the properties of telephone-wire lines is a formidable undertaking; we have been lucky enough to have received detailed information on telephone-line modeling through the offices of Ronald Sonderegger of DCEC and Captain Lemon of the Department of Defense. Armed with this information, Seneff[7] has succeeded in implementing a real-time simulation on the DVT. This allows us, by concatenating two DVTs, to do real-time listening of LPC with telephone input. Later portions of this report describe this experimental configuration in more detail.

REFERENCES

1. J. Makhoul and R. Viswanathan, "Quantization Properties of Transmission Parameters in Linear Predictive Systems," Report No. 2800, Bolt, Beranek and Newman (April 1974).
2. S. Seneff, private communication.
3. J.W. Forgie, private communication.
4. R. McAulay, private communication.
5. Semiannual Technical Summary, Information Processing Techniques Program, Lincoln Laboratory, M.I.T. (31 December 1975), DDC AD-B010167-L.
6. J. L. Flanagan, Speech Analysis, Synthesis and Perception (Springer-Verlag, Berlin, 1972).
7. S. Seneff, "A Real-Time Digital Telephone Simulation on the Lincoln Digital Voice Terminal," Technical Note 1975-65, Lincoln Laboratory, M.I.T. (30 December 1975), DDC AD-A021409/8.

IV. THE TELEPHONE-LINE SIMULATOR

A. OVERVIEW-DESCRIPTION OF A TELEPHONE CHANNEL

In a typical telephone channel,[1] the signal is first filtered, then modulated up to some carrier frequency, transmitted with multiplexing through a series of cables and repeaters, and finally filtered and demodulated at the receiver. Various distortions are introduced into the signal at all parts of the system. The filters pass only energy between 300 and 3000 Hz, and introduce both amplitude and phase distortion in the pass band. The modulation-demodulation process introduces quadrature distortion whenever there is an offset between the modulating and demodulating frequencies. Crosstalk is a result of both frequency and space division multiplexing, whenever the filtering and/or shielding are inadequate. In traveling down the cable, the signal becomes attenuated; therefore, the repeaters are necessary to amplify it at various points in the transmission lines. The repeaters are a source of both Gaussian noise and nonlinear distortion. When a signal is switched from one carrier to another, a sudden change in the phase relationship is introduced as well as transient (impulse) noise. Another source of impulse noise is lightning and corona-type discharges.

B. SINGLE-SIDEBAND (SSB) MODULATION

Typically, voice signals are frequency-multiplexed over a telephone line using SSB suppressed-carrier (SSBSC) amplitude modulation,[2,3] as shown in Fig. IV-1(a-d). In order to set the stage for our subsequent discussion of phase distortion, SSBSC is reviewed in this section.

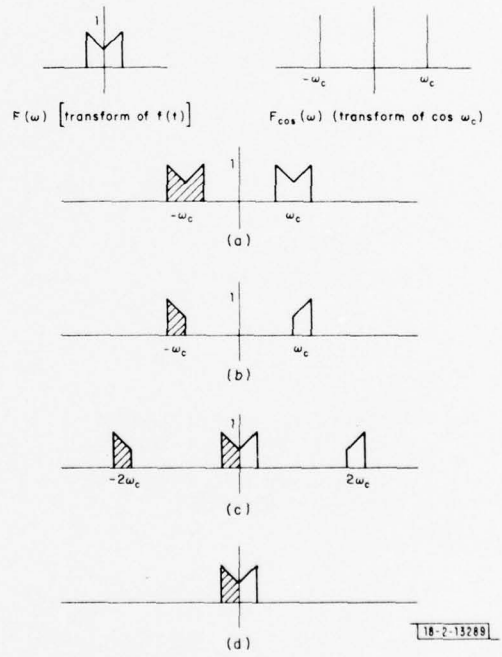


Fig. IV-1. SSB modulation-demodulation technique. (a) Modulated up; (b) bandpass-filtered to exclude frequencies below ω_c; (c) modulated down at receiver; (d) low-pass-filtered to restore $F(\omega)$.

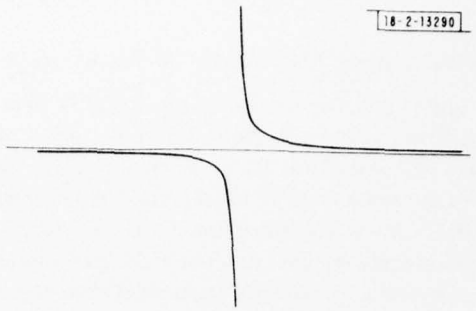


Fig. IV-2. Hilbert transform of an impulse.

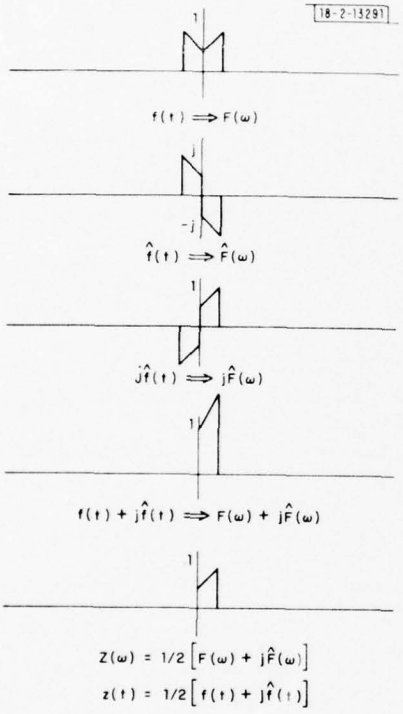


Fig. IV-3. Analytic signal z(t) expressed as a function of real signal f(t) and its Hilbert transform $\hat{f}(t)$. (See text for discussion.)

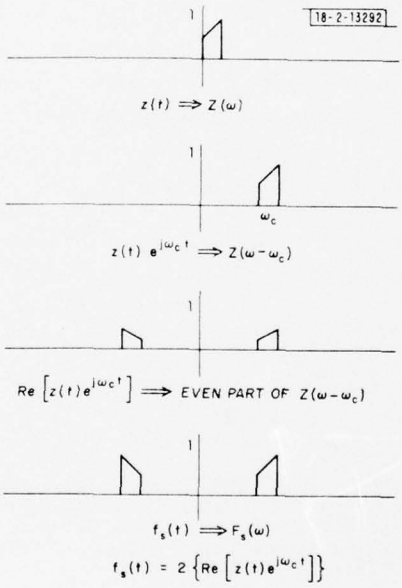


Fig. IV-4. SSB signal $f_S(t)$ expressed as a function of analytic signal z(t).

The signal is first bandpass-filtered from 300 to 3000 Hz and then modulated up by a carrier cosine whose frequency ω_c is a multiple of 4000 Hz. The resulting signal is then bandpass-filtered to eliminate any energy below ω_c. Since the original signal was real, this entails no loss of information. The signal is then frequency-multiplexed with several other signals at different carrier frequencies (also multiples of 4 kHz) and is detected at the receiver by means of a bandpass filter from $\pm\omega_c$ to $\pm(\omega_c + 4000$ Hz), followed by demodulation using a cosine also at frequency ω_c. However, no attempt is made to assure that the modulating and demodulating cosine waves are in phase with each other. In fact, the modulating and demodulating frequencies are rarely exactly the same, and any slight difference in frequency will cause a gradual drift into and out of phase.

When the modulating and demodulating cosines are exactly 90° out of phase, the resulting received signal is the quadrature component, or Hilbert transform,[4] of the original signal. The quadrature component can be understood from several different viewpoints. If one represents the original signal f(t) in terms of its Fourier expansion, i.e., as a sum of cosines at different frequencies with differing amplitudes and phases, then the Hilbert transform $\hat{f}(t)$ could be constructed by converting all the cosines to sines and adding them up. An impulse, for example, is a sum of cosines all in phase at $\Theta = 0$, at that point in time when the impulse occurred. If all these cosines were converted to sines, every function would be zero at the time of the impulse, all would be negative immediately before it, and all would be positive immediately after. Some distance away on either side, the sines would be at random phase with respect to one another, and hence the net sum at any point would be 0. The result is the function shown in Fig. IV-2.

One way to acquire the Hilbert transform is to convolve the signal with the function of Fig. IV-2. Another way is to take the Fourier transform, multiply positive frequency by $-j$ and negative frequency by $+j$ and take the inverse Fourier transform. It is clear that such a filter would convert a cosine at any frequency ω into a sine, since the transform of a cosine is two positive real impulses at $\pm\omega$ and the transform of a sine is a negative imaginary impulse at $+\omega$, and a positive imaginary impulse at $-\omega$ · $[(e^{j\omega} - e^{-j\omega})/2j = \sin\omega]$.

Supposing one had available a complex signal z(t) whose spectrum was zero for negative frequency and identical to the spectrum of a given real signal f(t) for positive frequency. According to the previous discussion, such a function could be generated as follows:

$$z(t) = 1/2\,[f(t) + j\hat{f}(t)] \qquad \text{(Fig. IV-3)} \quad .$$

If one were to multiply z(t) by $e^{j\omega_c t}$, the transform of the resulting function would be simply $Z(\omega)$ translated up by a frequency ω_c, i.e., $Z(\omega - \omega_c)$. Now, were one to take the real part of this signal and determine its transform the result would be the even part of $Z(\omega - \omega_c)$ which (Fig. IV-4) is clearly the same as what would be obtained by multiplying f(t) by $\cos\omega_c t$ and filtering out energy below ω_c. Remembering that the even part of a transform is the transform of the real part of the function, we have:

$$f_s(t) = 2\,\mathrm{Re}\left[z(t)\, e^{j\omega_c t}\right]$$

$$= \mathrm{Re}\left\{[f(t) + j\hat{f}(t)]\, e^{j\omega_c t}\right\}$$

$$= [f(t)\cos\omega_c t - \hat{f}(t)\sin\omega_c t] \quad .$$

Now it should be straightforward to explain the effect of a phase difference between the modulating and demodulating carriers:

$$f_o(t) = [f(t) \cos \omega_c t - \hat{f}(t) \sin \omega_c t] \cos (\omega_c t + \varphi) \qquad \text{(Low Pass)}$$

$$= f(t) \cos \omega_c t \cos (\omega_c t + \varphi) - \hat{f}(t) \sin \omega_c t \cos (\omega_c t + \varphi) \qquad \text{(Low Pass)}$$

$$= f(t) \cos \omega_c t \cos \omega_c t \cos \varphi - f(t) \cos \omega_c t \sin \omega_c t \sin \varphi$$

$$- \hat{f}(t) \sin \omega_c t \cos \omega_c t \cos \varphi + \hat{f}(t) \sin \omega_c t \sin \omega_c t \sin \varphi \qquad \text{(Low Pass)} \quad .$$

Recalling that

$$\cos^2 \theta = 1/2(1 + \cos 2\theta)$$

$$\sin^2 \theta = 1/2(1 - \cos 2\theta)$$

$$\sin\theta \cos\theta = 1/2(\sin 2\theta)$$

and that the $\sin 2\theta$, $\cos 2\theta$ terms will all be eliminated by low-pass filtering at the output, we have finally

$$f_o(t) = 1/2[f(t) \cos \varphi + \hat{f}(t) \sin \varphi] \quad .$$

When $\varphi = 90°$, $f_o(t) = 1/2\hat{f}(t)$, which is to say that only the quadrature term is received.

C. DISTURBANCES IN THE TRANSMISSION SYSTEM

1. Quadrature Distortion

Armed with a knowledge of the mathematics of SSB carrier systems, it is now simple to understand the various phase distortions that occur in the telephone systems. If there is a frequency difference $\Delta\omega$ between the modulating and demodulating waves, we would have:

$$f_o(t) = [f(t) \cos \omega_c t - \hat{f}(t) \sin \omega_c t] \cos [(\omega_c + \Delta\omega) t] \qquad \text{(Low Pass)} \quad .$$

The contribution to the phase difference φ by the frequency offset ω is a function of time:

$$\varphi_o = (\Delta\omega) t$$

which accounts for a slow drift into and out of phase.

Phase jitter is the term used to describe the interference of a sinusoidal noise (frequently 60 cycles) in the phase of a carrier. Sixty-cycle interference shows up everywhere in the system but, in general, is harmless because of the 300-Hz lower limit of the filters. However, 60-cycle interference in the generation of a cosine causes a phase distortion of the cosine waveform which appears as a very-low-frequency modulation (Fig. IV-5). The phase jitter introduces

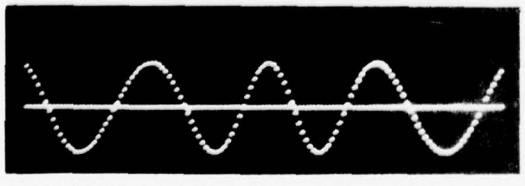


Fig. IV-5. Cosine wave at 312 cycles/sec distorted by phase jitter at 60 cycles/sec, 30° peak-to-peak.

an additive component to the phase which can be expressed as

$$\varphi_j = A_j \cos \omega_j t$$

where A_j is the amplitude of the jitter in degrees, and ω_j is usually 60 cycles (50 cycles in Europe).

The final phase effect is phase hits, or sudden large shifts in phase, which occur as a result of switching two carrier supplies not in phase. Sometimes a phase-coherent detector corrects the situation some time later. On other occasions, the original phase relationship is never restored.

The net phase relationship is a combination of the three effects:

$$\varphi = \varphi_o + \varphi_j + \varphi_h$$

and the final output signal is

$$f_o(t) = f(t) \cos \varphi + \hat{f}(t) \sin \varphi \quad .$$

2. Filtering

A major part of the distortion in a telephone signal has to do with the frequency-response and phase-delay characteristics of the filter used to assure that the signal remains in band. Theoretically, in multiplexing at 4000-Hz intervals, one could pass all frequencies from 0 to 4000 Hz and avoid crosstalk. In practice, filters never cut off sharply, and so a healthy compromise is to pass the signal from about 300 to 3000 Hz. In addition, there tends to be considerable noise interference at low frequencies (for example, 60-cycle hum) so that it is advantageous from a SNR argument to remove low frequencies. The characteristics of the filter in the pass band represent a trade-off between cost and response. In particular, not much attention was paid to phase-delay distortion, as the ear is insensitive to phase.

3. Nonlinear Distortion

In the process of traveling down a coaxial cable, a signal's energy is gradually attenuated due to losses in the transmission line and it is therefore necessary to amplify the signal at certain points in the transmission path to assure an adequate level at the receiver. The devices which achieve the amplification are called repeaters, and are generally made up of two-port amplifiers. The voltage-transfer characteristics of a generalized two-port are shown in Fig. IV-6.

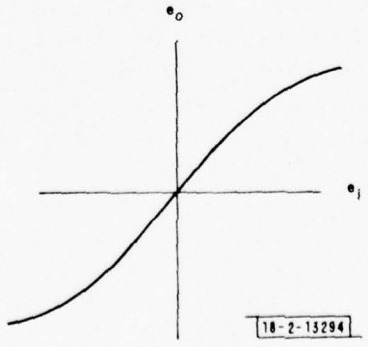


Fig. IV-6. Nonlinear voltage transfer characteristics of two-port.

According to a power series expansion, the transfer characteristics can be expressed as:

$$e_o = a_1 e_i + a_2 e_i^2 + a_3 e_i^3 + \cdots$$

where a_1 is the gain, and the other a's are the nonlinear distortion coefficients.

The sources of the nonlinear distortions are several, and much has been written on the subject of repeater design (see, for example, Ref. 1, pp. 396-421). The nonlinearities in a repeater are, in general, frequency-dependent, and the calculation of the precise response for a given system design is quite complicated.

4. Gaussian and Impulse Noise

Gaussian noise is an unavoidable side product of networks and devices. Two common types of noise in a circuit are thermal and shot noise (see Ref. 1, pp. 151-164). Arguing from the central limit theorem, both can be assumed to be Gaussian and white at the source. Thermal noise is present in any conductor (for example, a resistor) and is due to the thermal interaction between free electrons and vibrating ions. Its available power is directly proportional to the product of the bandwidth of the system and the temperature of the source. Shot noise is due to the discrete nature of electron flow and is present in most active devices (transistors and diodes). Its amplitude is proportional to the square root of the current, and therefore is dependent on signal level.

The components of the repeaters are a major source of the Gaussian noise. By the time the noise reaches the receiver, it is no longer white because it has been shaped by the filters at demodulation.

In addition to Gaussian noise, there is occasionally a burst of noise on a line whose amplitude far exceeds the average noise level of the system. Such noise is referred to as "impulse hits" and is generally caused by switching transients in central offices or from corona-type discharges (electrical discharges in the air surrounding a high potential line) that occur along a repeated line.

5. Echo and Crosstalk

To be complete in a report on telephone-channel characteristics, one must include at least a brief discussion of echo and crosstalk, even though these two effects were not included in the digital simulation. An echo may be produced whenever there is an impedance discontinuity. It is most commonly caused by a return of a talker's signal through the channel to which he is listening. For short distances, the effect is indistinguishable from side tone and is therefore not annoying. Echo-suppressor circuitry has been added to long-distance lines to attenuate the signal on the return path when the received signal amplitude is high. The result is that when both people talk only one is heard, and sometimes the beginnings of words are clipped. There is, however, a "tone-disabler" section of the echo-suppressor circuit which permits the mechanism to be shut off when data are being transmitted.

Crosstalk is caused by coupling losses between two active circuits. Transmittance crosstalk occurs because of inadequate design of modulators and filters in frequency multiplexing. Coupling crosstalk is caused by electromagnetic coupling between two physically isolated circuits. The result, of course, is that the listener hears another conversation in the background, which may or may not be intelligible.

D. MEASURING THE TELEPHONE CHANNEL

Before the telephone channel could be simulated, it was necessary to make extensive measurements on thousands of lines in order to determine concrete values to be used in the various components of the simulation. From the study data set, histograms were compiled for all the various parameters, such as frequency offset and level of Gaussian noise, for Continental U.S. (Conus) and European voice- and data-grade lines. Table IV-1 is a chart of the resulting numbers for the various degradations (excepting filter frequency response).[5] The term "mid" is used to refer to the 50th percentile on the histogram, and "poor" refers to the 90th percentile; i.e., 90 percent of the Conus voice-grade channels measured were better than "Conus poor voice."

TABLE IV-1

IMPAIRMENTS OF THE EIGHT TELEPHONE CHANNELS SIMULATED

Channel Simulated	Phase Hits	Frequency Offset (Hz)	Phase Jitter	Harmonic Distortion (dBmc)	Gaussian Noise (dBmc)	Impulse Noise
Conus poor voice	45/15 min. at 17°	1	15° p-p, 60 Hz	−20	−40	None
Conus mid voice	15/15 min. at 17°	0	11.6° p-p, 60 Hz	−25	−46	None
Conus poor data	45/15 min. at 17°	1	14° p-p, 60 Hz	−20	−37	None
Conus mid data	5/15 min. at 17°	0	11° p-p, 60 Hz	−28	−45	None
European poor voice	405/15 min. at 42°	7	35° p-p, 50 Hz	−37	−39	225/15 min., 74 dBrnc
European mid voice	45/15 min. at 32°	3.5	26° p-p, 50 Hz	−43	−45	25/15 min., 74 dBrnc
European poor data	135/15 min. at 42°	6	35° p-p, 50 Hz	−37	−39	225/15 min., 74 dBrnc
European mid data	135/15 min. at 22°	2.7	18° p-p, 50 Hz	−43	−45	25/15 min., 74 dBrnc

Gaussian noise and nonlinear distortion were measured in units of dBmc, meaning decibels relative to 1 milliwatt, using a special C-message weighting curve (see Ref. 1, pp. 31-34) shown in Fig. IV-7. This frequency-weighting curve was determined empirically by means of several subjective listener tests, and is intended to reflect the amount of listener annoyance for noises

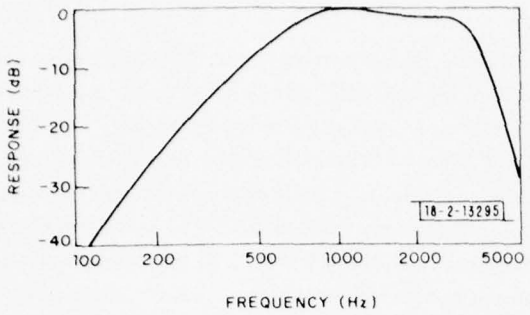


Fig. IV-7. C-message frequency-weighting function.

at different frequencies. The unit of measurement used for impulse noise is dBrnc, where "rn" stands for reference noise which is −90 dBm, or 10^{-12} W at 1000 Hz.

In addition to the C-message weighting curve, the device used by the telephone company to measure Gaussian noise has a built-in mechanism for attenuating impulse noise of a sufficiently short duration. The argument again is one of ear sensitivity, as the human ear does not fully perceive the power in noises of less than 200-msec duration.

The telephone company has available another device, called an impulse counter (see Ref. 1, p. 174), for the purposes of measuring the amount of impulse noise present, as such noise is far more destructive than Gaussian noise in introducing errors in the transmission of bits through a modem and a channel. The counter consists of a weighting network, a rectifier, a threshold detector, and a counter of events above threshold. Several impulse counters can be used simultaneously at different thresholds to obtain information about the distribution of the magnitudes of the impulses.

To measure the nonlinear distortion (see Ref. 1, pp. 238-239), a pure tone at a fixed level is transmitted, and the amount of energy received at the second and third harmonics is measured. From these values, one can deduce the coefficients a_2 and a_3 of the squared and cubed terms:

$$\cos^2\theta = 1/2(1 + \underline{\cos 2\theta})$$

$$\cos^3\theta = \cos\theta \cos 2\theta = 1/2(\cos\theta + \cos\theta \cos 2\theta)$$

$$= 1/2 \cos\theta + 1/4(\underline{\cos 3\theta} + \cos\theta) \quad .$$

Because of this relationship between nonlinearities and harmonics, nonlinear distortion is often referred to as harmonic distortion.

E. SIMPLIFICATIONS USED IN THE SIMULATION

The parameters of the simulation were set so as to emulate, as closely as possible, the results found from the statistical study. In the case of certain measures, such as frequency offset and phase jitter, this was a straightforward process. However, in the simulation of Gaussian noise, impulse hits, phase hits, and nonlinear distortion, certain (sometimes gross) simplifications were used.

The numbers used for Gaussian noise, impulse noise, and nonlinear distortion in the simulation were determined empirically by adjusting constants until the desired value was obtained at the output by the appropriate measuring device (described in Sec. D above).

Gaussian white noise generated at various devices in the transmission line is no longer white by the time it reaches the receiver because of the filter transfer characteristics. In the simulation, Gaussian noise is added at the input, so that it will be shaped by the filter of the system. The level of the Gaussian noise in the telephone system is dependent upon the amplitude of the input signal, whereas, in the simulation, noise is kept at a fixed level.

Impulse and phase hits are simulated to occur at fixed intervals and at fixed duration and level, so as to correspond in frequency of occurrence and average amplitude to the given telephone-channel characteristics.

Nonlinear distortion is simulated by squaring and cubing the final output of the system. The coefficients a_2 and a_3 remain fixed and are equal, even though in the actual telephone system these two coefficients would, in general, be dependent on both amplitude and frequency. The appropriate gain for the squared and cubed terms was of course determined so as to match the desired meter measurement. This was done by sending a pure tone at 700 Hz through the digital distorter and measuring the output of a notched filter (to remove 700 Hz) in dBrnc.

F. IMPLEMENTATION ON THE LDVT

The LDVT is a small high-speed computer which has proven capable of realizing in real time a variety of algorithms for low bit-rate digital-speech transmission[6] (from 2400 to 16,000 bps). The computer consists of a 1024-word program memory Mp with a cycle time of 55 nsec, a 512-word data memory Md, and a 2048-word outboard memory Mx, an input/output device from which data are accessible in a few instruction cycles. The LDVT has a very simple instruction set and is quite convenient to program.

The telephone-channel simulator represents the first attempt to use the LDVT for something other than a vocoder. A block diagram of the complete simulation is given in Fig. IV-8. The input speech is low-pass filtered to remove energy above 5 kHz, and sampled at 10 kHz using a 12-bit A/D converter. Impulse and Gaussian noise are generated digitally and added to the input samples.

The frequency response and delay characteristics of the telephone filter and the Hilbert transform to obtain the quadrature component are both realized by means of high-speed-convolution techniques.[7] After a 256-point FFT of the input samples (256 points is adequate as long as the combined impulse response of the filter and Hilbert transform is less than 12.8 msec), negative frequency is zeroed out and positive frequency is complex-multiplied by the FFT of the desired impulse response. A 256-point IFFT now yields the filtered speech f(t) in the real data buffer, and the filtered quadrature component $\hat{f}(t)$ in the imaginary data buffer (refer to the previous discussion on SSB modulation in Sec. B above). Only the second half of the data is good, because of circular convolution; therefore, an overlap of 128 samples is necessary between frames.

The phase φ is computed as the sum of the contributions of phase jitter, frequency offset, and phase hits. The output of the "modulator-demodulator" complex is then computed as:

$$g(n) = f(n) \cos\varphi + \hat{f}(n) \sin\varphi \quad .$$

The final step in the simulation is to subtract from g(n) the nonlinear distortion term:

$$\hat{s}(n) = g(n) - D\,[g^2(n) + g^3(n)] \quad .$$

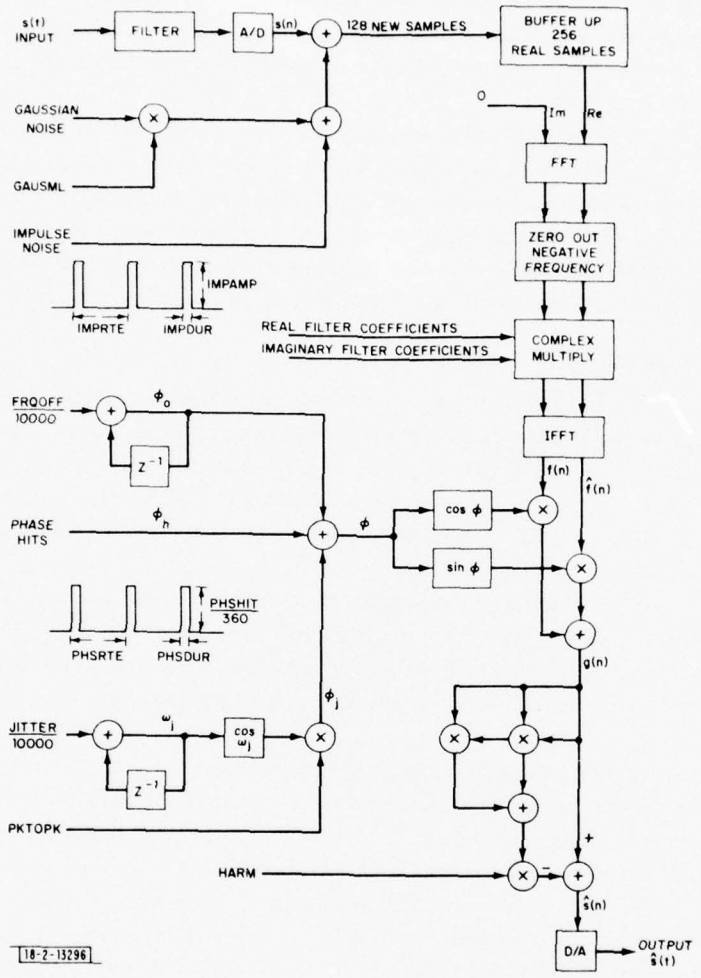


Fig. IV-8. Block diagram of telephone-channel simulator.

```
                        DATA FOR CONUS POOR VOICE                    -2-13297

012403   003720    PHSRTE    2000.    :PERIOD BETWEEN PHASE HITS IN .01. sec UNITS
012404   000001    PHSDUR    1        :DURATION OF PHASE HITS IN .01. sec UNITS
012405   000001    FRQOFF    1        :FREQUENCY OFFSET IN HZ
012406   000017    PKTOPK    15.      :DEGREES PEAK TO PEAK OF JITTER
012407   000074    JITTER    60.      :HZ
012410   000021    PHSHIT    17.      :DEGREES
012411   050475    HARM      .6347.   :HARMONIC DISTORTION FACTOR
012412   001320    GAUSML    .022.    :NEW SIGMA = GAUSML*.25.
012413   000000    IMPRTE    0        :PERIOD BETWEEN IMPULSE HITS IN .01. sec UNITS
012414   000000    IMPAMP    .0.      :AMPLITUDE OF IMPULSE HITS
012415   000001    IMPDUR    1        :DURATION OF IMPULSE HITS IN .01. sec UNITS
012416   000000    PKTFRC    .0.      :FRACTIONAL PART OF PKTOPK
012417   000000    OFFFRC    .0.      :FRACTIONAL PART OF FRQOFF
```

Fig. IV-9. Parameters of synthesizer reside in locations 403 to 417 of data memory Md.

There is a special buffer in Md (data memory) which is set aside as the parameters of the system (excepting filter frequency response) and which can be modified under user control. Figure IV-9 is a copy of that section of the program, including the appropriate settings for Conus poor voice. The same mnemonics are used in Fig. IV-8, where it should be clear how the various parameters are being used.

Table IV-2 gives the values used for the various parameters in the simulation of the eight channels. The values for the parameters FRQOFF, PKTOPK, JITTER, PHSHIT, PKTFRC, and OFFFRC are simply copied over from Table IV-1. The values for PHSRTE and IMPRTE are determined as the reciprocal of No./15 min., and converted from units of minutes to units of 0.01 sec. The parameters PHSDUR and IMPDUR are left unspecified in Table IV-1, and are set arbitrarily to 0.01 sec in Table IV-2 for all channels. The parameters GAUSML, HARM, and IMPAMP are dimensionless fractions determined empirically so as to read the correct meter reading at the output in units of dBmc or dBrnc.

The only other information that is needed to simulate each of the channels is the frequency-domain characteristics of the telephone filter. Tables were given for each channel of 256 real and 256 imaginary frequency components of the desired filter, spaced by 20 Hz, spanning 0 to 5 kHz. Since the DVT implementation used a 256- rather than 512-point FFT, alternate samples were ignored in entering the tables into the computer. It should be noted here that it would have been essentially impossible to implement a 512-point FFT in the DVT due to its current memory limitations, but that the 12.8-msec window appears to be adequate for the filters simulated.

The main body of the program is the FFT-IFFT computation which requires several buffers in Mx (outboard memory), two 128-point buffers in Md, and a complex interchange of data between Mx and Md. The program takes advantage of certain characteristics of the data to reduce both time and memory requirements. By keeping the FFT size down to 128, one avoids the necessity of referencing Mx in the inner loop, which would greatly increase the time required. Therefore, the forward FFT is realized (since the input data are real) by packing even-numbered samples in the real Md buffer MDREAL, and odd-numbered samples in the imaginary Md buffer MDIMAG, and doing a 128-point FFT followed by even-odd separation and a final stage of the 256-point FFT. The first stage of the inverse FFT is nonexistent, since the second half of the data is zero, and therefore it can be conveniently split into two 128-point FFTs, with the first

TABLE IV-2

VALUES USED FOR THE PARAMETERS OF THE SIMULATOR TO CORRESPOND TO THE DATA IN TABLE IV-1

	Conus Poor Voice	Conus Mid Voice	Conus Poor Data	Conus Mid Data	European Poor Voice	European Mid Voice	European Poor Data	European Mid Data
PHSRTE (0.01-sec units)	2000	6000	2000	18,000	222	2000	667	667
PHSDUR (0.01-sec units)	1*	1	1	1	1	1	1	1
FRQOFF (Hz)	1	0	1	0	7	3	6	2
PKTOPK (deg)	15	11	14	11	35	26	35	18
JITTER (Hz)	60	60	60	60	50	50	50	50
PHSHIT (deg)	17	17	17	17	42	32	42	22
HARM	0.6348	0.3455	0.6348	0.2563	0.0928	0.0464	0.0928	0.0464
GAUSML	0.022	0.0085	0.0275	0.0098	0.0198	0.0082	0.0195	0.0082
IMPRTE (0.01-sec units)	0	0	0	0	400	3600	400	3600
IMPAMP	0	0	0	0	0.11	0.11	0.11	0.11
IMPDUR (0.01-sec units)	1*	1	1	1	1	1	1	1
PKTFRC (deg)	0	0.6	0	0	0	0	0	0
OFFFRC (Hz)	0	0	0	0	0	0.5	0	0.7

* PHSDUR & IMPDUR are variables of the system which, however, were fixed at 0.01 sec for these simulations since there was no information given.

one yielding the even-numbered output samples and the second one (by beginning each stage with the coefficient index set at half its increment instead of at zero) yielding the odd-numbered output samples.

Because of this even-, odd-numbered sorting at both the output and input, it is convenient to store all buffers of speech in Mx in this peculiar fashion of even-numbered samples in one buffer and odd-numbered samples in another. The buffers required in Mx are listed in Table IV-2. The even-numbered input samples are stored in EVIO and the odd-numbered in ODIO. At each A/D interrupt, the next sample is fetched out of either EVIO or ODIO alternately and sent to the D/A converters, and the new sample from the A/D converter is written over the same place in Mx. At the beginning of a new frame, the most recent 128 samples from the A/D converter have filled up EVIO and ODIO, and so these buffers now become EVNEW and ODNEW, respectively. At the same time, the former EVNEW and ODNEW become EVOLD and ODOLD, and the former EVOLD and ODOLD whose contents had been written over by $\hat{s}(n)$ in the course of the previous frame, become EVIO and ODIO, ready to be sent to the D/A converter.

The first step in a new frame is to fill MDREAL with 128 even-numbered new samples from EVOLD/EVNEW and to fill MDIMAG with the odd-numbered samples from ODOLD/ODNEW. Now a 128-point normal to bit-reversed decimation in frequency FFT followed by bit-reversed even-odd separation and a final stage of the 256-point FFT can all be realized in place in Md. The resulting data (the first 128 samples of the 256-point FFT) are then complex-multiplied by the bit-reversed filter coefficients (which have been stored in Mx). The resulting filtered spectrum is then saved in Mx: the real data are stored over ODOLD/EVOLD, and the imaginary data are saved in a 128-point buffer, MXIMAG. Of the data in Md, a 128-point IFFT at this point yields the even-numbered f(n) samples in MDREAL and the even-numbered $\hat{f}(n)$ (quadrature component) samples in MDIMAG.

Only the second half of the data is good, because of circular convolution, and these 64 even-numbered f(n) and even-numbered $\hat{f}(n)$ are saved in Mx. Meanwhile, the original real and imaginary outputs of the forward FFT filtered spectrum are fetched back from Mx and another 128-point IFFT, beginning each stage with the coefficient index set at half its increment instead of at 0, yields the odd-numbered samples of f(n) and of $\hat{f}(n)$ in Md. Again, the first 64 of each are garbage. The 64 even-numbered samples of f(n) and of $\hat{f}(n)$ can now be fetched back from Mx and stored over the 64 garbage points in each of MDREAL and MDIMAG. Now, all the pertinent data reside in Md and all that is left is the introduction of quadrature distortion and nonlinear distortion:

$$g(n) = f(n) \cos\varphi + \hat{f}(n) \sin\varphi$$

$$\hat{s}(n) = g(n) - D\,[g^2(n) + g^3(n)] \quad .$$

This can be done conveniently at this time, and the resulting $\hat{s}(n)$ evens can be stored over EVOLD, and the $\hat{s}(n)$ odds over ODOLD.

When the interrupt routine has been serviced 128 times, resulting in 64 new samples in each of EVIO and ODIO, it is time to swap buffer pointers again, leaving $\hat{s}(n)$ evens in EVIO and $\hat{s}(n)$ odds in ODIO, as desired, and we have come full circle.

The assurance of adequate accuracy without overflow in the FFT-IFFT complex at first posed some difficulty, as a check for overflow and correction in the inner loop proves very costly in terms of time. The scheme finally decided upon was:

(1) Shift the input buffer of the forward FFT in Md as far left as possible as a block, so that the largest sample does not overflow.

(2) Scale down the data as a block by 1/2 at each stage of the inner loop of the forward FFT.

(3) Scale the output data in Md from the forward FFT as far left as possible.

(4a) If the data have been scaled up by more than 2^N, scale them down by 1/2 at each stage of the IFFT and scale them down the remaining amount at the end.

(4b) If the data have been scaled up by less than 2^N, scale them down by 1/2 at each stage of the IFFT until such time as no more scaling is needed.

The remainder of the program is reasonably straightforward. Gaussian and impulse noise are added to a new s(n) in the interrupt routine, as soon as it comes in from the A/D converter and before it is sent to Mx. Gaussian noise is simulated by first generating a 9-bit pseudo-random number with a flat distribution. Eight bits are used as a pointer into a table of the mid-points of 256 equal areas over the positive half of a Gaussian distribution, and the ninth bit determines the sign. The numbers in the table were chosen such that $\sigma = 0.25$, and values (when fetched from the table) are multiplied by the appropriate constant GAUSML to obtain the desired σ. Impulse noise is created by adding to each s(n) a signal of height IMPAMP for the duration of time IMPDUR, and at a periodic rate IMPRTE.

The phase φ and the components φ_o, φ_h, and φ_j are all stored in the computer as fractions, with 1.0 corresponding to 360°. φ_o is updated at each s(n) by adding FRQOFF/10,000, since an offset of 1 Hz would be 360° in 10,000 samples. Likewise, ω_j is updated by adding JITTER/10,000 at each s(n). The cosine of ω_j is multiplied by PKTOPK/360 (expressing degrees peak-to-peak as a fraction) to obtain φ_j. Phase hits are simulated by adding PHSHIT/360 to φ for those samples s(n) during the interval PHSDUR, and spaced by the time PHSRTE. The three values $\cos\omega_j$, $\cos\varphi$, and $\sin\varphi$ are determined by table lookup from Mx using the same first-quadrant 64-point cosine table as is used for the FFTs, plus an additional interpolation angle of 90°/128 whose cosine and sine are stored in Md. From these it is possible to determine the cosine and sine of any angle from 0° to 360° to the accuracy of 90/128° using the formulas:

$$\cos(a + b) = \cos a \cos b - \sin a \sin b$$

and

$$\sin(a + b) = \cos a \sin b + \cos b \sin a \quad .$$

Harmonic distortion is added just before the final adjustment of the data from block floating to fixed point so as to gain bits in the cubing and squaring of the data. The terms $g^2(n)$ and $g^3(n)$ are added together, multiplied by the parameter HARM, and subtracted from g(n) to obtain $\hat{s}(n)$ which, when converted back to fixed point, is the final output of the system.

A somewhat intuitive feel for the effects of phase jitter and frequency offset can be gained by referring to Fig. IV-10(a-c). This figure shows a time exposure of the impulse response of the telephone filter for Conus poor voice with (a) all the other distortions removed from the system, (b) only phase jitter added, and (c) only frequency offset added. It can be noted that, with

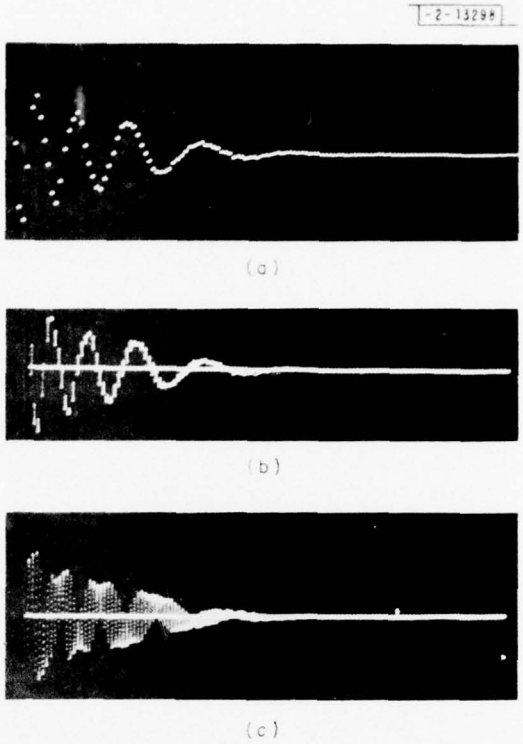


Fig. IV-10. Time exposure of unit sample response of filter for Conus poor voice line. (a) Unit sample response; (b) same as in (a) but with phase jitter at 17° peak-to-peak, 60 Hz; (c) same as in (a) but with frequency offset between transmitter cosine and receiver cosine of 1 Hz.

a phase-jitter interference, the samples "jitter" about a focal point, whereas with a frequency offset there is a smooth motion of the wave as it appears to continually pass through and disappear.

G. DVT TIMES AND SPACE

Table IV-3 is a list of the various subroutines in the LDVT in their approximate order of occurrence. Included for each subroutine are the amount of program memory required and the amount of time required for its execution. The total time needed per frame is slightly more than half the time available, and the program uses up 77 percent of Mp.

There are only 512 locations in Md, of which half are needed for the 128-point FFT real and imaginary buffers. Fortunately, no other large buffers are needed by the program, so that an additional 125 locations were needed for the various parameters, temporaries, and variables, leaving one-fourth of Md unused.

As for outboard memory Mx, 256 locations are needed for the filter coefficients, 256 for the Gaussian table, 128 for the cosine tables, and 4×128 for the various speech buffers, for a total of 1152 or 56 percent of Mx (Table IV-4).

TABLE IV-3

LIST OF PROGRAM SUBROUTINES, AND TIME AND PROGRAM MEMORY REQUIRED IN TELEPHONE-CHANNEL SIMULATOR

Subroutine	Function	Program Memory	Time (msec)
GTSPCH	Fetch input speech from Mx, store in Md.	29	0.078
SCLUP	Scale data as block left maximally, twice.	39	0.422
FFTFWD	128-point forward FFT.	137	1.2
EOLL	Bit-reversed even-odd separation. Last loop of 256-point FFT.	109	0.28
FLTMUL	Complex multiply by filter coefficients.	29	0.28
SVEBUF	Save filtered spectrum in Mx.	19	0.08
IFFT1	128-point IFFT to obtain even-numbered samples.	94	1.20
GTBUF	Save f(n), $\hat{f}(n)$, even-numbered, in Mx. Fetch filtered spectrum from Mx.	44	0.12
IFFT2	128-point modified IFFT to obtain odd-numbered samples.	– (shared)	1.20
FTCEVN	Fetch f(n), $\hat{f}(n)$, evens, back from Mx.	14	0.39
PHASIT	Compute phase, compute $g(n) = f(n) \cos \phi + \hat{f}(n) \sin \phi$. Add nonlinear distortion.	121	0.96
ADJSCL	Change $\hat{s}(n)$ from block floating to fixed point.	29	0.046
STRSPC	Store $\hat{s}(n)$ in Mx.	16	0.039
ADINT	Interrupt routine. Service A/D-D/A. Add Gaussian and impulse noise to s(n). Update counters for phase hits and impulse hits.	109	0.46
	Total	789 = 77 percent	6.755 msec = 53 percent

TABLE IV-4

BUFFERS IN OUTBOARD MEMORY Mx NEEDED FOR TELEPHONE SIMULATION

Buffer	Allocation	Buffer Size
EVIO	Even-numbered in-out samples	64
ODIO	Odd-numbered in-out samples	64
EVNEW	Most recent 64 even-numbered samples of s(n)	64
ODNEW	Most recent 64 odd-numbered samples of s(n)	64
EVOLD	Previous 64 even-numbered s(n) } also as Mx Real	64
ODOLD	Previous 64 odd-numbered s(n) } also as Mx Real	64
MXIMAG	Temporary storage of imaginary FFT output	128
GAUS	Midpoints of 256 equal areas of positive half of Gaussian	256
FLTR	Bit-reversed filter coefficients (128 Real, 128 Imaginary)	256
SINE	Sine of 0 to $\pi/2$ for FFTs and for computation of cosine ϕ	64
RVSINE	Bit-reversed sine of 0 to $\pi/2$ for last stage of forward FFT	64
	Total	1152 = 56 percent

REFERENCES

1. Transmission Systems for Communications, 4th Edition, Bell Telephone Laboratories (1970).
2. J. M. Wozencraft and M. Jacobs, Principles of Communication Engineering (Wiley, New York, 1967), pp. 504-508.
3. M. Schwartz, Information Transmission Modulation and Noise (McGraw-Hill, New York, 1970), pp. 218-228.
4. B. Gold, A. V. Oppenheim, and C. M. Rader, Theory and Implementation of the Discrete Hilbert Transform, Symposium on Computer Processing in Communications, Polytechnic Institute of Brooklyn (Polytechnic Press, 1970), pp. 235-250.
5. Captain R. Lemon, private communication.
6. P. E. Blankenship et al., "The Lincoln Digital Voice Terminal System," Technical Note 1975-53, Lincoln Laboratory, M.I.T. (25 August 1975), DDC AD-A017569/5.
7. B. Gold and L. R. Rabiner, Theory and Application of Digital Signal Processing (Prentice-Hall, New York, 1975).

V. THE HARMONIC PITCH DETECTOR

A. INTRODUCTION

All pitch detectors can be placed in one of two categories – time domain and frequency domain. Time-domain pitch detectors deal directly with the speech waveform, and as such are relatively fast, since very little preprocessing of the signal is required. Most frequency-domain detectors require an abundance of time and memory storage to obtain the spectral information over a sufficiently long time window and with adequate spectral resolution. These methods are therefore often not realizable in real-time vocoder implementations, or are only realizable at the cost of excessive quantization of the pitch.

Pitch detection is generally good when the input signal is intact and noise-free. However, distortions, filters, and noise tend to obscure the pitch information and cause most pitch detectors to break down, sometimes severely. Since in the real world the signal is often corrupted, we felt that an algorithm designed to be robust against degradations would be a significant new contribution.

We were particularly interested in coping with degradations caused by (1) passage of the speech through the public telephone system prior to pitch detection, and (2) acoustically coupled noise backgrounds. From a previous effort,[1] we had the capability to simulate in real time the filtering, phase distortion, phase jitter, and nonlinear distortion effects of a telephone system. In addition, we had available test material wherein the noise background of a large jet airplane was incorporated into the recording.

The algorithm thus developed is a frequency-domain technique which, however, restricts itself to a selected portion of the frequency band below 1100 Hz. Digital-signal-processing tricks were used to obtain the desired spectral region with minimal computation time. Pitch is determined from spacing between peaks in this region, using an iterative method. The buzz-hiss decision makes use of none of the standard indicators such as energy ratios and zero crossing density, as these parameters are highly susceptible to noise and distortion. Instead, continuity of the pitch track is the only parameter used to determine voicing, other than a very conservative silence threshold. The algorithm has been incorporated into a real-time linear-prediction vocoder implemented on the Lincoln Digital Voice Terminal (LDVT).[2]

B. PREPROCESSING

In order to obtain an accurate pitch estimate from the spectral information, it is necessary to begin with a spectrum with good frequency resolution, but one which spans a sufficient block of the frequency space for there to be at least two harmonics present over the range available. Since a pitch value of 350 Hz is not unreasonable for a female, the spectral region to be analyzed must be at least 700 Hz wide. Within this range, one can arbitrarily choose an FFT size to yield the desired frequency spacing between samples at the expense of computer time. We decided to compute a 128-point FFT to yield a spectrum spanning 840 Hz with a resulting frequency spacing between samples of 6.6 Hz, which appears to be adequate resolution for our purposes.

Since the pitch information must be extracted from precisely the 840-Hz region chosen, it is expedient to carefully select that region which is most likely to yield robust harmonics. Since the telephone filter removes the signal below about 300 Hz, one need not waste space on the low

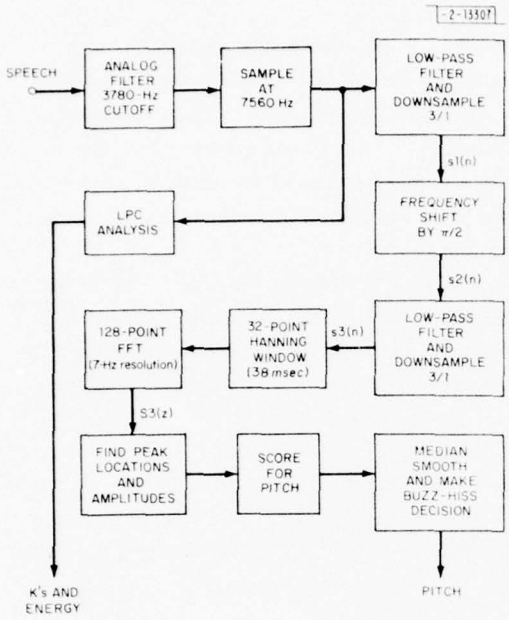
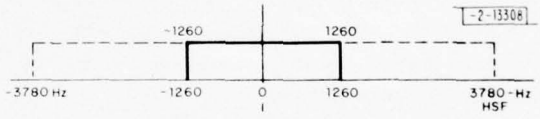
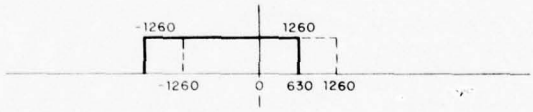


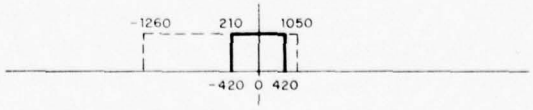
Fig. V-1. Block diagram of harmonic pitch detector.



(1) LP FILTER AND DOWNSAMPLE 3/1



(2) SHIFT SPECTRUM: MUL EACH s(n) BY $e^{j\pi n/2}$



(3) LP FILTER AND DOWNSAMPLE 3/1

Fig. V-2. Preprocessing of speech waveform to obtain downsampled signal with desired spectral information.

end of the frequency spectrum. However, as one advances to increasingly higher frequencies, the spectrum becomes more and more ragged, and the harmonics increasingly difficult to extract.

The region selected was 210 to 1050 Hz. These particular numbers were arrived at in large part because of appropriate tricks that could be used to extract precisely this piece of the spectrum. The original speech waveform was analog filtered and sampled at 130-μsec intervals, yielding a signal containing frequencies up to 3780 Hz, which could be used as input both to the linear-prediction analysis and to the pitch extractor.

The first step in the pitch extraction is to filter the speech down to 1260 Hz and downsample, discarding two-out-of-every-three samples (Fig. V-1). For this purpose, a finite impulse response (FIR) filter seemed to be a good choice. Since FIR filters have only zeros, one need compute outputs only at the downsampled rate, which in our case represents a three-to-one savings in time. Furthermore, FIR filters are implementable using charge-coupled devices (CCDs), a potentially fast and inexpensive computational source.

We now have a waveform s1(n) which contains information from −1260 to +1260 Hz. We know that, since the waveform is real, the negative frequency information is redundant. Digital-signal-processing theory tells us that if we multiply each sample of the waveform s1(n) by $e^{j\omega n}$, we will cause the spectrum to be rotated by ω in the z-plane. By choosing $\omega = 90°$, we cause the spectrum to be rotated such that 1260/2 = 630 Hz is at the origin (Fig. V-2). Now a second pass of this complex s2(n) through the same filter, with 3-to-1 downsampling again, will yield a complex waveform s3(n) containing frequencies up to 1260/3 = 420 Hz. However, because of the rotated spectrum, 630 Hz in the original waveform corresponds to 0 Hz in s2(n), and thus our doubly downsampled complex waveform contains the information from 630 − 420 Hz to 630 + 420 Hz in the original speech, which is the desired spectral region.

Choosing $\omega = 90°$ has certain advantages in terms of speed. Multiplication by $e^{j\omega n}$ involves only data transfer rather than complex multiplies, since the sine and cosine of multiples of 90° are always either ±1 or 0. Furthermore, as a consequence, each sample of s2(n) is either purely real or purely imaginary. One can therefore use simple tricks in the implementation of the FIR filter so that filtering of this complex waveform takes essentially no more time than would filtering a real waveform.

Pitch detection generally requires a long time window of speech in order to assure at least two periods of a low pitched voice. Fortunately, the doubly downsampled signal consists of samples which are spaced by 132 × 3 × 3 μsec, or 1.188 msec. Only 32 samples of this waveform are required to yield 38 msec of data, a time window that is sufficient to encompass two periods for pitches of up to 19 msec, or 53 Hz, a very deep male voice.

The 32 most recent samples of s3(n) are windowed using a standard Hanning window and then filled out with zeros to make a 128-point input buffer for the FFT. Because three-fourths of the input samples are zero, the FFT computation time can be reduced by essentially skipping the first two stages. The resulting spectrum contains the information in the original speech signal from 210 to 1050 Hz, as desired, and is ready, after the computation of the magnitude spectrum from real and imaginary components, to be processed for harmonic detection.

C. PEAK PICKING ALGORITHM

The self-normalized magnitude spectrum obtained from the windowed s3(n) is generally a very smooth function with peaks only at the harmonics of the pitch. The peaks are of unequal

size, the larger ones showing up at the resonance of the first formant. In the case of the phoneme /i/ for example, a vowel with an extremely low F1 frequency, the first harmonic is generally very large compared with all the others. On the other hand, the back vowel /a/ generally has a more graceful bulge in the high end of the spectrum, with the largest peak near 800 Hz or so (Fig. V-3).

The variability in size of peaks would not be a problem if there were never any spurious peaks. Unfortunately, such is not the case, for the speech waveform never behaves in any guaranteed fashion. A common problem is the presence of subharmonic peaks in the spectrum half way between the true harmonics, possibly caused by irregularities in the laryngeal excitation. These are nearly always smaller than their neighbors, but they may very well not be smaller than other true harmonics not at the formant resonance. Thus, a simple measure of distance between peaks above a fixed threshold may yield a better score for a pitch choice in hertz of half the true value. A further serious problem with telephone speech is that the carrier cosine often contains 60-Hz interference which shows up as 60-Hz modulation of the speech waveform. The consequence of such interference is spurious peaks on either side of a large peak, 60 Hz away. These are often larger than true harmonics not at the formant resonance (Fig. V-4).

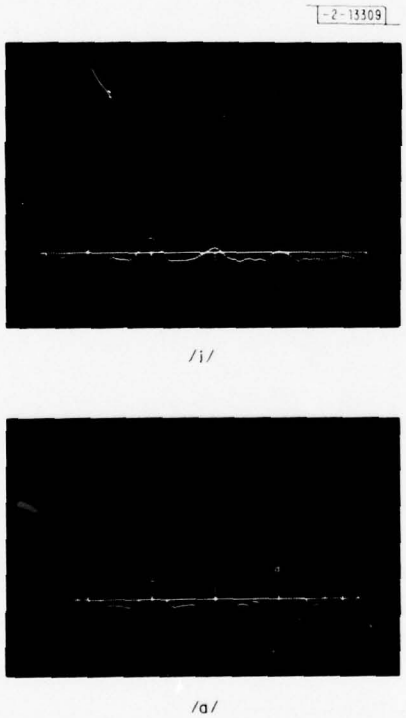


Fig. V-3. Typical first formant region spectra for vowels /i/ and /a/.

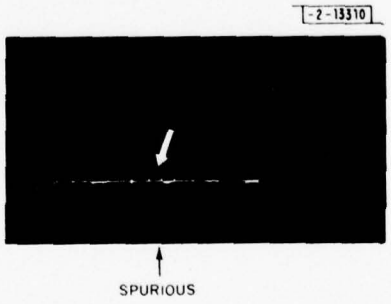


Fig. V-4. Introduction of spurious peak in spectrum as consequence of 60-Hz phase jitter.

Another fact which increases the difficulty of pitch detection is the wide variability in the number of peaks to expect to find. For a high-pitched female voice, there are often only two peaks which should even be considered, and the pitch is the distance between them. For an 80-Hz male voice, on the other hand, one expects to find at least 10 peaks at the harmonics of the pitch. An algorithm has to recognize the fact that there may be only two valid peaks, yet most of the time it should consider far more than two peaks in making a decision.

The algorithm described here uses an iterative technique which begins by considering only the two largest peaks. It then adds each peak in turn, from largest to smallest, and after the addition of each new peak determines a new list of potential pitches as the distance between adjacent peaks under consideration. Such a technique results in a built-in weighting mechanism, whereby the largest peak is included in every iteration, but the smallest only in the last. The final decision algorithm determines the pitch from a list which includes all the estimates from each iteration.

The first step in extracting the pitch is to find all peaks in the spectrum and to eliminate from consideration those which are judged to be spurious. For each peak, an amplitude and a frequency location are determined. The location is defined simply as the frequency at which the actual peak occurs. The amplitude is defined not as the magnitude of the sample at the peak, but rather as the "area under the hump." That is to say, the amplitude of a given peak is the non-normalized sum of the amplitudes of all the samples from the previous valley to the following valley. In the event that the sum overflows 16 bits, it is clamped at +1. This choice of definition was found to effect a better separation between true peaks and spurious peaks than would a simple amplitude at the peak.

Peaks are eliminated from consideration if they are too small and/or too close to a neighboring peak. Specifically, a peak is removed if its location is within 6 samples (40 Hz) of a larger neighboring peak. A peak which is more than 6 but fewer than 10 samples away from its nearest neighbor is removed if its amplitude is less than 1/2 the amplitude of the near neighbor.

The peaks that remain after the elimination step are given a rank order according to size. At the first iteration, a single pitch estimate is entered into a table of potential pitch estimates, defined as the distance between the two largest peaks. At the second iteration, the third largest peak is added to the list of peaks under consideration and two new pitch estimates are added to the table, defined as the distance between adjacent peaks, among the three under consideration. At each subsequent i^{th} iteration, the largest peak among those remaining is added to the list of candidate peaks and i new pitch estimates are added to the growing list of estimates, defined again as distance between adjacent peaks (Fig. V-5).

Pitch estimates are always added to the table in order, with the smallest at the beginning of the table. After each iteration, a score is computed for the maximum number of consecutive "equal" pitch estimates in the table. (Equal is defined as within 14 Hz of the succeeding entry in the table.)

As soon as there are at least 6 "equal" estimates, the average value for the "equal" entries is defined as the pitch (in Hz). If there are fewer than 6 "equal" estimates, then the algorithm continues with the next iteration until the size of the next available leftover peak is less than 1/10 the size of the largest peak, or until a maximum of 7 peaks has been exhausted. If either of these conditions is met, the algorithm exits in spite of an inadequate score, and chooses as the pitch value the average of the longest string of "equal" estimates. In the case of a tie between two strings, the one with the larger pitch estimate is arbitrarily defined to be the pitch.

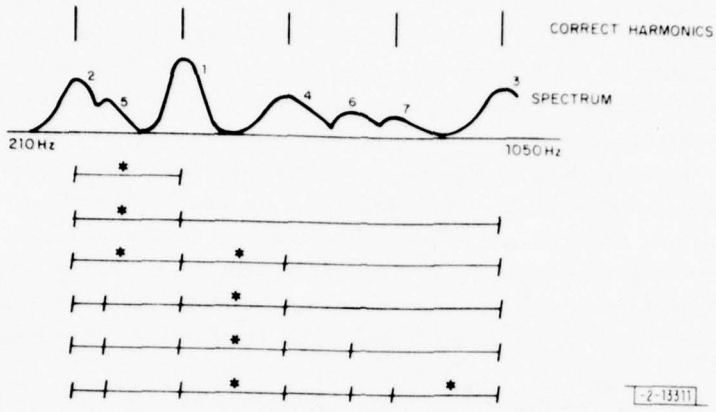


Fig. V-5. Illustration of iterative scoring algorithm under artificially adverse conditions.

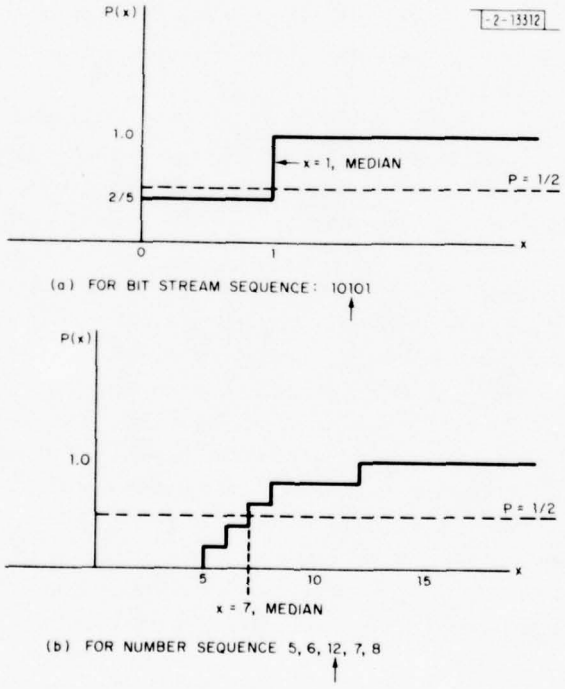


Fig. V-6. Median smoothing filter (a) for bit stream, and (b) for function.

This harmonic-detection algorithm is run twice per 20-msec frame on spectra of data spaced by 10-msec intervals. The output is thus an oversampled, unsmoothed pitch contour, and the final step in the processing is to make the buzz-hiss decision and decide a single pitch value for each frame. For this purpose, the pitch contour is passed through a 3-point followed by a 5-point median smoothing filter[3] [Fig. V-6(a-b)].

The buzz-hiss decision is made almost exclusively on the basis of the smoothness of the pitch contour. Since the only true feature distinguishing voiced from unvoiced speech is the presence of pitch pulses, and since the linguistic and acoustic constraints on the pitch make it highly unlikely for a true pitch value to change dramatically in the course of a 10-msec interval, one can expect that in voiced regions the pitch will change little from sample to sample. In unvoiced regions, on the other hand, there is little reason to expect the algorithm to arrive at anything other than random values for the pitch choice. The only other feature used by the buzz-hiss decision is an extremely conservative silence threshold on the doubly downsampled waveform s3(n).

Thus, the buzz-hiss decision operates as follows. If the energy in s3(n) is less than the silence threshold, consider the frame hiss and set the pitch equal to 0. If none of the three input samples to the 3-point median smoothing filter are "equal" (where equal is here defined as within 33 Hz of each other), consider the output of the median smoother to be 0 (hiss). Finally, if no more than 2 of the 5 ordered input samples to the 5-point median smoother are "equal" (this time within 20 Hz of each other), consider the output of the 5-point smoother to be 0 (hiss) (Fig. V-7).

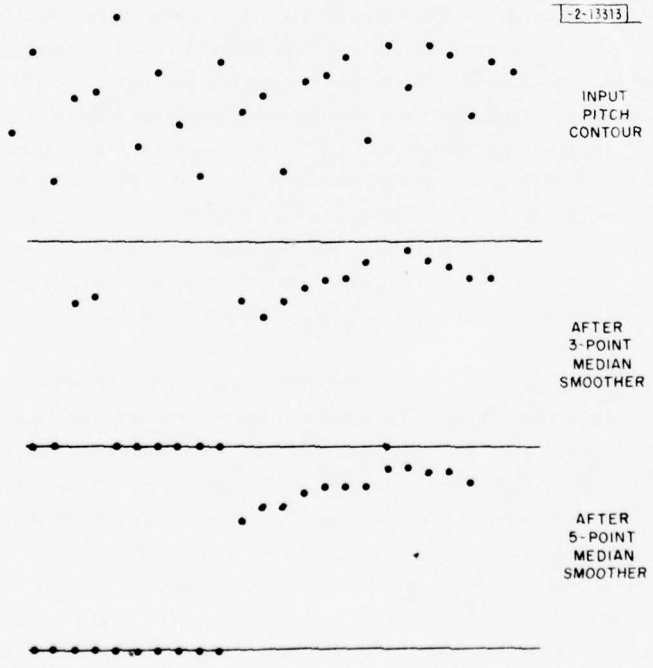


Fig. V-7. Illustration of median smoothing buzz-hiss algorithm under artificially adverse conditions.

This algorithm works surprisingly well for determining buzz-hiss. It depends upon a 10-msec rather than a 20-msec update of the pitch. Typical buzz-hiss indicators such as zero crossing density, R1/R0, and high-low energy ratios were avoided on purpose, because these are likely to be degraded as a consequence of filters, distortions, and noise to which the input speech may have been subjected.

D. LDVT IMPLEMENTATION

The algorithm as described above was incorporated into a real-time linear-prediction vocoder implemented on the LDVT, a 55-nsec instruction cycle microcomputer, with a standard instruction set, designed and built at Lincoln Laboratory. Memory size is a limiting factor with the LDVT, for it has only 2000 octal program and 1000 octal data memory locations. There is, however, a rapid access outboard memory containing 4000 octal locations from which both programs and data can be retrieved.

The pre-emphasized analog waveform was filtered and sampled at 132-μsec intervals, and a nonoverlapping buffer of 153 samples was accumulated for each 20-msec frame. These 153 samples were used as input both to the autocorrelator and to the first FIR filter, FIR1 (refer to Fig. V-1). The 51 output samples of FIR1 were complex multiplied by $e^{j\pi n/2}$ and processed again through the FIR filter, using certain tricks to handle the complex input data, to yield 17 new samples of s3(n). The 128-point FFT was computed twice per frame by moving along alternately by 9, then 8, samples of s3(n). The computation of the magnitude spectrum from the 32 most recent samples of s3(n), padded out with zeros to 128, completed the preprocessing.

For the postprocessing, a table of peak locations and a corresponding table of amplitudes were determined and arranged in descending order with respect to peak size. Following this step, the first two location entries were reordered and the difference between the two locations was entered as the first pitch estimate. Then the third entry in the location table was inserted in order and two new pitch estimates, defined as difference between adjacent entries, were added, also in order, to the growing pitch-estimate table. Now the three ordered entries in the estimate table could be scored for "equality" of adjacent elements, and an iteration is completed. At each i^{th} iteration, the i^{th} location is inserted in order, and i new pitch estimates are added in order to the estimate table. Processing is complete either when a peak of insufficient amplitude is encountered, or a score of greater than seven adjacent "equal" estimates is obtained. At this point, the mean value of the "equal" set is defined as the (unsmoothed) pitch.

An appreciation of the complexity of the algorithm can be gained from some numbers associated with the LDVT implementation. The total number of memory locations required for the entire pitch algorithm was 1425 decimal, divided about 50-50 between instructions and data. The amount of time consumed for the preprocessing (FIR filters and computation of magnitude spectrum) was 2.66 msec per 10-msec frame, or a little over a quarter of the time available. The time required for the postprocessing, or decision algorithm, was extremely variable and therefore difficult to determine, but a rough calculation indicates that it was insignificant compared with preprocessing time. For purposes of comparison, the total time requirement was roughly twice the amount required by the LDVT implementation of the Gold-Rabiner time-domain pitch detector.

E. RESULTS

The harmonic pitch detector, incorporated into a real-time 4000-bits/sec LPC vocoder, was evaluated subjectively by means of an AB comparison with the Gold-Rabiner time-domain detector,[4] incorporated into an otherwise identical vocoder. A system was developed on the UNIVAC 1219 facility whereby the two vocoders could be exchanged into the LDVT essentially instantaneously, while speech subjected to various distortions and corruptions was continuously being played. The listener could thus, because of the instantaneous juxtaposition, readily compare the quality of the speech produced using the frequency- and time-domain pitch detectors.

Input speech subjected to typical telephone-channel degradations was generated by means of a second LDVT containing a real-time digital telephone-channel simulator[1] (Fig. V-8). The parameters of the simulator were controlled at the console and thus the user could conveniently test the performance of the two pitch detectors, with increasing amounts of various corruptions. For example, if one wished to investigate the sensitivity of the two pitch detectors to Gaussian noise, one could set all parameters of the telephone simulator to zero except the Gaussian noise. The noise amplitude could then be slowly increased while the two pitch detectors were alternately loaded into the other LDVT.

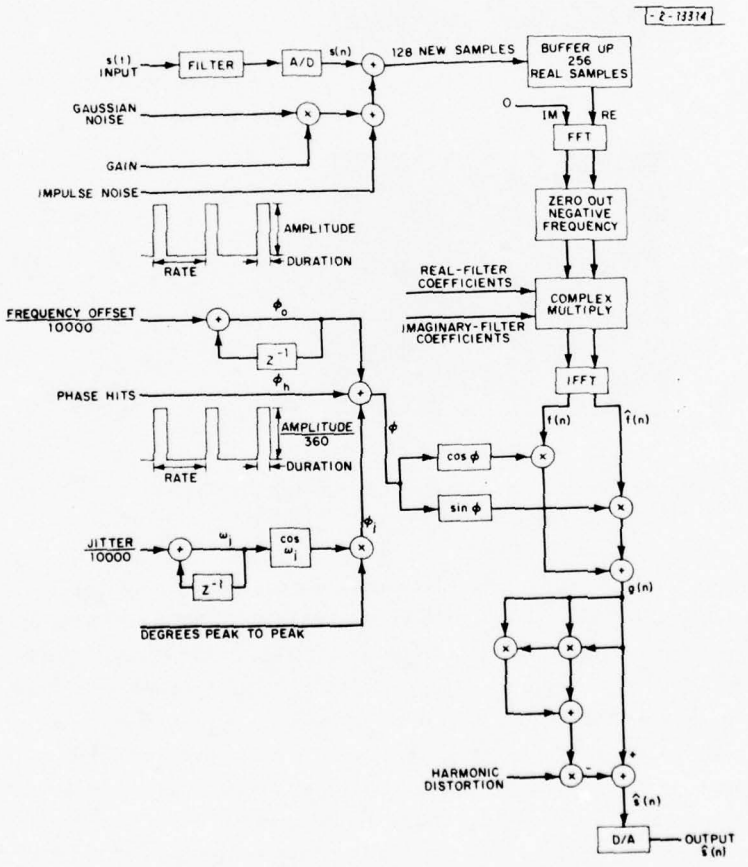


Fig. V-8. Block diagram of telephone-channel simulator used to test harmonic pitch detector performance.

Using this experimental setup, we were able to examine the relative sensitivity of the two pitch detectors to the various distortions in the telephone lines. The major source of breakdown in the Gold-Rabiner pitch detector is the telephone bandpass filter, which removes information below 300 Hz, attenuates the amplitude up to as much as 1000 Hz, and changes the phase relationship. Subjective listening tests show a substantial improvement in quality when the harmonic pitch detector is substituted for the time-domain detector, under conditions when only the telephone filter is present in the simulation. Figure V-9(a-b) shows an example where the periodicity is not evident in the waveform, but is well indicated in the spectrum, when the speech is processed through a typical telephone filter.

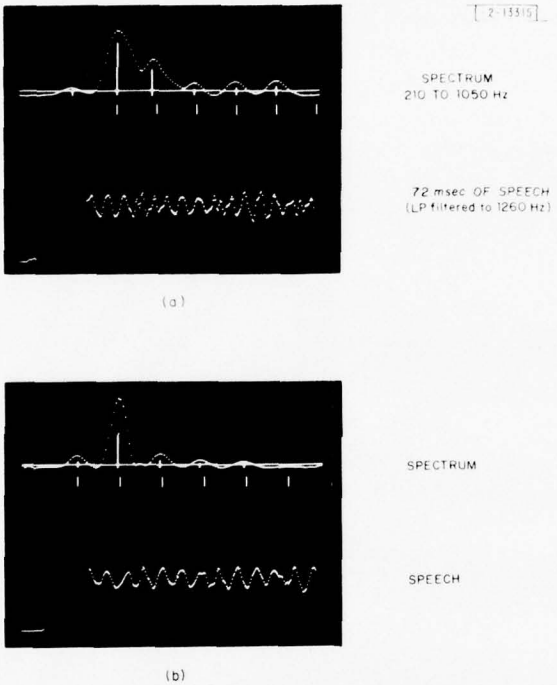


Fig. V-9. Waveform and spectrum of (a) telephone filtered speech, and (b) speech corrupted with jet-engine noise.

Other audible degradations in typical telephone lines are Gaussian noise (thermal noise and shot noise) and phase jitter. The latter is a low frequency modulation of the waveform as a consequence of (usually 60 Hz) interference in the generation of the carrier cosine. In some European lines, the 50-Hz jitter can be as high as 35-deg peak-to-peak amplitude, causing a peculiar granular quality and an echo effect in the speech.

As might be anticipated, both detectors were sensitive to Gaussian noise, although the breakdown as a consequence of Gaussian noise was not as great as might be expected. The Gold-Rabiner detector was far more sensitive to the telephone filter alone than to Gaussian noise alone, set at the level typically encountered in telephone lines (−40 dBmc). The two detectors were judged to be about equally sensitive to Gaussian noise.

Phase jitter contributes an additional degradation to the time-domain detector, particularly at the levels encountered in European lines. Included in the harmonic pitch detector decision algorithm is a step to suppress peaks too close to neighbors and of insufficient amplitude, which makes the detector less sensitive to phase jitter than the time-domain detector. At typical American line settings, phase jitter presents little problem to either detector.

The remaining parameters in the simulator, with the possible exception of harmonic distortion, seem to have little effect on pitch extraction, at the levels commonly found in the telephone system.

The two pitch detectors were also evaluated on certain other types of degraded speech. Specifically, speech in the presence of (1) helicopter noise, (2) noise in a large jet airplane, and (3) 60-Hz hum was processed through both vocoders, and the quality was compared. Helicopter noise was found to be concentrated in frequencies above 1000 Hz, and therefore caused only minor degradations in both detectors. Jet noise includes a large component in the low-frequency region (below 300 Hz) and therefore interferes rather severely with the time-domain pitch-extraction algorithm. The same is true, obviously, for 60-Hz hum, whose strongest component is at 60 Hz but which contains weaker harmonics at higher frequencies.

For both the 60-Hz hum and the jet-engine noise, the harmonic pitch detector performed substantially better than the time-domain detector. Even at levels of hum in which the time-domain detector completely broke down, choosing 60 Hz as the pitch, the harmonic detector came through with clear speech. Figure V-9(b) shows an example where the pitch information is obscure in the waveform but evident in the spectrum, when the speech is corrupted with large airplane jet noise.

For one specific kind of distortion in transmission channels, the harmonic detector can actually correct the distortion and improve the quality of the original speech. This is for the situation in which there happens to be a very large frequency offset between the transmitter and receiver carrier in a single side-band transmission system. In such a case, both positive and negative frequency are shifted in toward the origin by an amount equal to the offset, such that the original pitch harmonics are no longer harmonics. The subjective result is that the perceived pitch is wrong, and a small amplitude background hum is heard at the correct pitch. The harmonic pitch detector, since it does not depend upon the fundamental but only upon spacing between harmonics, can restore the original speaker's pitch in the synthesized speech, and remove the background hum. The formant frequencies are of course still shifted, but the formant shift is a second-order effect, perceptually.

REFERENCES

1. S. Seneff, "A Real-Time Digital Telephone Simulation on the Lincoln Digital Voice Terminal," Technical Note 1975-65, Lincoln Laboratory, M. I. T. (30 December 1975), DDC AD-A021409/8.
2. P. E. Blankenship et al., "The Lincoln Digital Voice Terminal System," Technical Note 1975-53, Lincoln Laboratory, M. I. T. (25 August 1975), DDC AD-A017569/5.
3. L. R. Rabiner, M. R. Sambur, and C. E. Schmidt, "Applications of a Nonlinear Smoothing Algorithm to Speech Processing," IEEE Trans. Acoust., Speech, and Signal Processing ASSP-23, 552 (1975).
4. B. Gold and L. R. Rabiner, "Parallel Processing Techniques for Estimating Pitch Periods of Speech in the Time Domain," J. Acoust. Soc. Am. 46, 442 (1969).

VI. OPTIMUM SPEECH CLASSIFICATION AND ADAPTIVE NOISE CANCELLATION

A. INTRODUCTION

There are a variety of applications in which it is necessary to be able to classify a given set of speech data as corresponding to voiced speech, unvoiced speech, or silence. For the synthesis of speech using Linear Predictive Coding (LPC) techniques,[1-4] for example, it is necessary that the speech signal be classified as voiced or unvoiced. This information is transmitted to the speech synthesizer along with coefficients that represent an all-pole linear filter model for the vocal tract. For voiced speech, the filter is excited by a periodic train of impulses, whereas a white-noise excitation is used when unvoiced speech is to be synthesized.

The ability to detect silence is of interest in digital communications in which channel capacity is at a premium.[5] By detecting intervals of silence, other data streams can be interleaved with the speech conversation, thereby maximizing the utilization of the available bandwidth. Another application of silence detection arises in conferencing situations.[5] By detecting when a set of speakers are silent, their lines can be disconnected from the superposition of inputs so that an enhancement of synthesizer input SNR can be obtained.

Solutions to the classification problem have, for the most part, been developed on an ad hoc basis in which an individual discriminant is proposed which seems to characterize, in one way or another, the attributes of the three possible speech events. In a recent paper, Atal and Rabiner[6] proposed an algorithm that simultaneously computes five of the most-significant discriminants and uses a hypothesis testing strategy to assign a given set of observations to one of the three speech classes.

With few exceptions, most notably the work of Atal and Rabiner, most of the speech research reported to date has dealt with a speech environment that has been carefully controlled in the sense that background noise and interference signals have been eliminated from the speech. It is generally known that the intelligibility of modern vocoders is seriously degraded when noise and interference signals are superimposed on the speech data.[5] Since there are many practical problems in which noise and interference arise, it is of interest to develop more general speech-processing techniques designed to eliminate the noise as much as possible.

In this section, it is assumed that the speech signals are corrupted by additive Gaussian noise that may or may not be white. The unvoiced-speech signal is modeled as a zero-mean Gaussian random process having a known covariance function. Voiced speech is modeled as a zero-mean Gaussian quasi-periodic random process. By using these models as a starting point, the classification problem is formulated as a statistical hypothesis test and is solved using statistical decision theory. Subject to the validity of the underlying speech models, the resulting signal-processing algorithm is optimum in the sense that the probability of a decision error is minimized. The advantage of this approach is that the discrimination criteria are synthesized from the model, rather than being selected on an ad hoc basis.

The classification problem is recognized as a Gauss-in-Gauss detection problem for which solutions have been cataloged by Van Trees.[7] The estimator-correlator structure was chosen since it led most naturally to a practical implementation. If pitch information is available, additional discrimination can be provided in the voiced-speech channel using a comb filter tuned to the most-recent estimate of the pitch.

The ability to detect the silent intervals (noise alone) means that the statistics of the clutter can be learned and used to implement adaptive Wiener filters to enhance the speech signals prior to coding. In this mode, the adaptive prefilter can be used as a preprocessor for any narrowband or wideband speech encoder.

An extensive experimental program was developed to evaluate the classifier in a variety of acoustic-noise environments including shipboard noise, office noise, helicopter noise, and noise in an airborne command post. The results for airborne-command-post noise are included in this section.

B. MODELS FOR SILENCE, UNVOICED, AND VOICED SPEECH

The basic problem of detecting the presence of silence, unvoiced speech, or voiced speech in a given set of data can be formulated as a statistical test for choosing one of the three hypotheses:

$$\begin{aligned} H_1&: \text{silence} && y(n) = w(n) \\ H_2&: \text{unvoiced} && y(n) = u(n) + w(n) \\ H_3&: \text{voiced} && y(n) = v(n) + w(n) \end{aligned} \qquad \text{(VI-1)}$$

where w(n), u(n), and v(n) represent the n^{th} sample of noise, unvoiced-speech, and voiced-speech waveforms, respectively. Based on a set of observations $y(1), y(2), \ldots, y(N)$, it is desired to develop a decision rule for determining which of the three hypotheses "best" characterizes the data set. This is the classification problem. In order to synthesize an optimum decision rule in the sense that a classification is made with minimum probability of error, it is necessary to develop statistical models that characterize the data for each of the three speech events.

To begin with, the interference will be assumed to consist of simply zero-mean white Gaussian noise. Once the detector structure has been analyzed and understood for this case, the generalization to nonwhite-noise spectra follows almost by inspection.

In order to derive the structure of the classifier, it suffices to model the unvoiced- and voiced-speech waveforms as sample functions of Gaussian random processes having zero means and covariance functions $R_u(k)$ and $R_v(k)$, respectively. In addition, voiced speech is assumed to be quasi-periodic in the sense that $R_v(k + T) = R_v(k)$, where T is the period of the process. This means that almost every sample function is periodic with period T (see Ref. 8).

The preceding discussion can be summarized succinctly by the following set of modeling equations. Under hypothesis H_i, the observed data set is given by:

$$y(n) = s_i(n) + w(n) \qquad i = 1, 2, 3 \qquad \text{(VI-2)}$$

where $s_1(n) = 0$ for silence, $s_2(n)$ is a Gaussian random process with mean zero and covariance $R_u(k)$ for unvoiced speech, and $s_3(n)$ is a zero-mean quasi-periodic Gaussian random process with covariance function $R_v(k)$ for voiced speech. In all cases, the noise term w(n) represents a zero-mean Gaussian white-noise random process having the correlation function $R_w(k) = \sigma^2\delta(k)$.

C. THE OPTIMUM CLASSIFIER AGAINST WHITE NOISE

The optimum classifier processes the raw-speech data $y(1), y(2), \ldots, y(N)$ in such a way that a decision is made with minimum probability of error on whether the given interval of signal

should be classified as voiced speech, unvoiced speech, or silence. Using statistical decision theory, the minimum probability error decision rule is:

"Declare hypothesis H_i to be true if and only if the a posteriori probability that H_i is true conditioned on the observation set $y(1), y(2), \ldots y(N)$ is largest," i.e.,

$$p\,[H_i \,|\, y(N), \ldots, y(1)] = \max_{k=1,2,3} p\,[H_k \,|\, y(N), \ldots, y(1)] \quad .$$

Signal-processing configurations of the likelihood-ratio test have been documented by Van Trees.[7] For the special case of ternary hypotheses, zero means, and stationary random processes, the test is implemented by computing three sufficient statistics denoted by ℓ_i (i = 1,2,3). The first component of the i^{th} statistic is

$$\ell_{yi} = \sum_{n=1}^{N} y(n)\, \hat{s}_i(n) \qquad \text{(VI-3)}$$

where $\hat{s}_i(n)$ is the linear least-squares unrealizable estimate of the i^{th} signal $s_i(n)$. The bias component of the i^{th} sufficient statistic is

$$\ell_{Bi} = -\frac{T}{2} \int_{-\infty}^{\infty} \ell n \left[1 + \frac{G_i(f)}{N_o/2}\right] df \qquad i = 1, 2, 3 \qquad \text{(VI-4)}$$

where $T = N/F_s$ is the observation time of the process, F_s is the sampling rate, $G_i(f)$ is the power spectrum of the i^{th} random process, and $N_o/2$ is the two-sided white-noise spectral density. The complete i^{th} sufficient statistic is

$$\ell_i = \ell_{yi} + \ell_{Bi} \qquad i = 1, 2, 3 \qquad \text{(VI-5)}$$

and the test consists of choosing the largest of

$$\ell_i + \ell n P_i \qquad i = 1, 2, 3 \qquad \text{(VI-6)}$$

where P_i is the a priori probability that hypothesis H_i is true. The goal now is to use the gross attributes of speech signals to simplify the computations involved in implementing the likelihood-ratio test.

Under hypothesis H_1, which corresponds to silence, the anticipated signal is $s_1(n) = 0$. Therefore, $\hat{s}_1(n) = 0$ whence $\ell_{y_1} = 0$, $\ell_{B_1} = 0$, and $\ell_1 = \ell n P_1$. The likelihood-ratio test reduces to computing only two statistics:

$$\ell_2 = \ell_{y_2} + \ell_{B_2} + \ell n P_2 - \ell n P_1 \qquad \text{(VI-7a)}$$

$$\ell_3 = \ell_{y_3} + \ell_{B_3} + \ell n P_3 - \ell n P_1 \qquad \text{(VI-7b)}$$

in which only ℓ_{y_2} and ℓ_{y_3} involve the raw data, ℓ_{B_2} and ℓ_{B_3} being fixed biases reflecting the average energy in the ensembles of unvoiced- and voiced-speech sounds. Letting

$$\lambda_u = -\ell_{B_2} - \ell n P_2 + \ell n P_1 \qquad \text{(VI-8a)}$$

$$\lambda_v = -\ell_{B_3} - \ell n P_3 + \ell n P_1 \qquad \text{(VI-8b)}$$

$$\lambda_{uv} = -\ell_{B_2} + \ell_{B_3} - \ell n P_2 + \ell n P_3 \qquad \text{(VI-8c)}$$

the classification rule reduces to the following:

$$\text{If } \ell_{y_2} \leqslant \lambda_u \quad \text{and} \quad \ell_{y_3} \leqslant \lambda_v \qquad \text{declare silence} \tag{VI-9a}$$

$$\text{If } \ell_{y_2} > \lambda_u \quad \text{or} \quad \ell_{y_3} > \lambda_v \quad \text{and} \quad \ell_{y_2} - \ell_{y_3} \geqslant \lambda_{uv} \qquad \text{declare unvoiced speech} \tag{VI-9b}$$

$$\text{If } \ell_{y_2} > \lambda_u \quad \text{or} \quad \ell_{y_3} > \lambda_v \quad \text{and} \quad \ell_{y_2} - \ell_{y_3} < \lambda_{uv} \qquad \text{declare voiced speech} \tag{VI-9c}$$

In order to simplify the test further, it is noted from Eq. (VI-4) that the bias terms ℓ_{B_2} and ℓ_{B_3} are related to the energy in the ensemble of unvoiced- and voiced-speech sample functions. If a global average is taken, the voiced-speech spectrum will have significantly more energy than that of unvoiced speech, which would contribute a negative bias in favor of the unvoiced-speech hypothesis. Using this bias would be valid if voiced speech were truly stationary. In fact, however, not only do the spectral properties change from frame to frame, but more importantly the amplitude undergoes a slowly increasing and decreasing modulation at the beginning and ending of a voiced sound. Since 10- to 20-msec frames of speech represent the data base upon which a classification is to be made, then from a sample function point of view the energy in a frame of unvoiced speech or a frame of voiced speech could be comparable. The inclusion of the ensemble average energy bias term would therefore incorrectly favor unvoiced speech. Therefore, the bias terms ℓ_{B_2} and ℓ_{B_3} must be assumed to be equal. Under this condition, the thresholds reduce to

$$\lambda_u = -\ell_B - \ell n P_2 + \ell n P_1 \tag{VI-10a}$$

$$\lambda_v = -\ell_B - \ell n P_3 + \ell n P_1 \tag{VI-10b}$$

$$\lambda_{uv} = -\ell n P_2 + \ell n P_3 \tag{VI-10c}$$

where $\ell_B = \ell_{B_2} = \ell_{B_3}$ represents an unknown bias term related to the a priori knowledge of the energy in the unvoiced- and voiced-speech signals.

Although the Bayesian detection theory demands that the bias term and a priori probabilities be calculated, a more practical method for determining the thresholds would be to train the system against noise and then choose those values that keep the false-alarm rate at a value consistent with the system objectives. For example, a much greater penalty is paid for failing to detect speech than falsely classifying noise as speech. Therefore, the thresholds most likely should be set close to the 1-sigma values of ℓ_{y_2} and ℓ_{y_3} obtained during the noise training phase. This strategy is ideal for self-adaptive tracking of the noise statistics should they be nonstationary. The voicing threshold λ_{uv} is most reasonably approximated by zero when the SNR is large or the noise is white. When this is not the case, this threshold can also be trained to the 1-sigma value of $\ell_{y_2} - \ell_{y_3}$.

As a result of the preceding analysis, the only statistics that must be calculated at each frame time are the correlations

$$\ell_{y_i} = \sum_{n=1}^{N} y(n)\, \hat{s}_i(n) \qquad i = 2, 3 \tag{VI-11}$$

where y(n) is the raw-speech-plus-noise data, and $\hat{s}_i(n)$ is the linear least-squares unrealizable estimate of $s_i(n)$ given that hypothesis H_i is true. Since the unvoiced- and voiced-speech waveforms are quasi-stationary, the filter that results in $\hat{s}_i(n)$ given that $y(n) = s_i(n) + w(n)$ has the transfer function

$$H_i(f) = \frac{G_i(f)}{G_i(f) + N_o/2} \quad . \tag{VI-12}$$

The filters defined by Eq. (VI-12) obtain enhanced discrimination against noise by passing only those frequencies where the signal power is substantially larger than the noise power. Enhanced voice-unvoiced discrimination depends on the implicit orthogonality of the two random processes, as reflected by the degree to which the spectral densities are correlated. Both these detection statistics can be improved by capitalizing on the quasi-periodic properties of voiced speech. If the voiced-speech process is periodic with period T, then the voiced-speech power spectrum is more accurately represented by

$$G_3(f) = C(f; T)\, G_v(f) \tag{VI-13}$$

where $G_v(f)$ represents the gross properties of the spectral envelope, and $C(f;T)$ is a comb filter reflecting the fine structure of the periodic spectrum. If the period is maintained for M periods, then

$$C(f;T) = \frac{1}{M} \frac{\sin(\pi M f/F)}{(\pi f/F)} \cdot \exp[j\pi(M-1)\, f/F] \tag{VI-14}$$

where $F = 1/T$ represents the pitch frequency. Not only does the comb filter enhance the voiced-speech-to-noise ratio, but it also increases the orthogonality of the voiced and unvoiced spectra. In order to exploit the additional discrimination implicit in the comb filter, it is necessary that the pitch period be known. A discussion of how the pitch is to be determined will be deferred to Sec. E below.

Subject to the assumptions that the envelopes of the unvoiced- and voiced-speech power spectra are known and that the pitch period for voiced speech can be estimated, then the optimum classifier can be implemented as shown in Fig. VI-1.

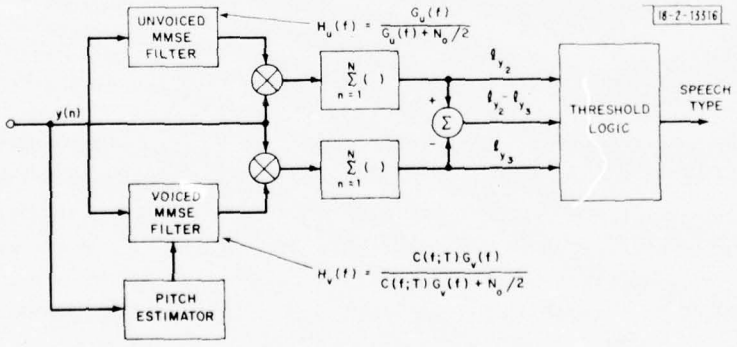


Fig. VI-1. Optimum speech classifier against white noise.

Of course, all this information is not available a priori and it will be necessary to introduce approximations to the filtering and estimation operations while maintaining the basic structure of the estimator-correlator receiver. This will be the goal of the next section.

D. PRACTICAL IMPLEMENTATION OF THE ESTIMATOR-CORRELATOR SPEECH CLASSIFIER

For voiced speech, the optimum minimum mean-squared error filter has the transfer function

$$H_v(f) = \frac{G_v(f)\, C(f;T)}{G_v(f)\, C(f;T) + N_o/2} \qquad \text{(VI-15)}$$

which passes those frequencies at which the signal power is substantially larger than the noise power and rejects all others. Certainly, the comb filter in the denominator contributes to the definition of those frequencies at which noise rejection should occur. However, in white noise approximately the same rejection performance can be obtained by a cascade combination of the comb filter and the least-squares filter designed on the basis of simply the spectral envelope. Therefore, the voiced-speech estimator filter is taken to be

$$H_v(f) = C(f;T) \cdot \frac{G_v(f)}{G_v(f) + N_o/2} \quad . \qquad \text{(VI-16)}$$

For unvoiced speech, the estimator filter is

$$H_u(f) = \frac{G_u(f)}{G_u(f) + N_o/2} \quad . \qquad \text{(VI-17)}$$

Setting i = 2 for unvoiced and i = 3 for voiced, the Wiener filters based on the spectral envelopes for both cases can be written as

$$H_i(z) = \sum_{k=-\infty}^{\infty} a_k^i z^{-k} \qquad \text{(VI-18)}$$

where the coefficients a_k^i satisfy the Wiener-Hopf equation

$$\sum_{k=-\infty}^{\infty} a_k^i \left[R_i(k-j) + \sigma^2\, \delta(k-j)\right] = R_i(j) \qquad -\infty < j < \infty \qquad \text{(VI-19)}$$

where $\sigma^2 = (N_o/2)\, F_s$ represents the energy in the noise process (F_s is the sampling rate), and where $R_2(k)$, $R_3(k)$ are the sampled data-correlation functions corresponding to the power spectra $G_u(f)$, $G_v(f)$, respectively. In practice, the correlation functions can be suitably truncated and then Eq. (VI-18) can be efficiently solved using the Levinson recursion.[9] Of course, the solution requires that the correlation functions for an ensemble of unvoiced- and voiced-speech sample functions be computed for a large class of utterances and a large class of speakers. In order to bootstrap the system, initial classification would have to be done manually, which would be extremely tedious and time consuming. In order to avoid this problem, a more practical and robust strategy is proposed based on the well-known global properties of unvoiced- and voiced-speech spectra and a close examination of the filtering operation defined in Eqs. (VI-16) and (VI-17).

The essence of the Wiener filter is to pass those frequencies at which the speech power is substantially larger than the noise power. As a good first approximation, it seems reasonable to approximate the Wiener filter by a passband filter that passes "most" of the energy in an unvoiced- or voiced-speech sound. For unvoiced speech, it can be assumed that "most" of the energy will be above 1000 Hz, while for voiced speech "most" of the energy will be below 2000 Hz. While restricting the estimator filters to these frequencies improves the detection SNR of unvoiced and voiced speech, of at least equal importance is the ability of the unvoiced filter to reject voiced speech and vice versa. Since the first formant of voiced speech is approximately 1000 Hz, then, if the cutoff of the unvoiced-speech filter is above 1250 Hz, most of the unvoiced-speech energy will pass through the filter while a large fraction of a voiced-speech signal will be attenuated. Similarly, if the cutoff of the voiced-speech signal is above 2000 Hz, then most of its energy will pass through the voiced filter, while a substantial fraction of an unvoiced-speech signal will be attenuated. From this point of view it can be seen that it is crucial that the input data to the classifier <u>not</u> be pre-emphasized since the higher formants of a voiced-speech signal would take on the attributes of an unvoiced-speech waveform at the expense of good classifier performance. Therefore, if pre-emphasis is to be used for speech analysis and synthesis, the data will have to undergo digital de-emphasis prior to speech classification.

On the basis of the preceding arguments, the Wiener filter for unvoiced speech will be approximated by a high-pass linear-phase digital filter whose cutoff frequency is below 1250 Hz. For voiced speech, a low-pass linear-phase digital filter having a cutoff frequency above 2000 Hz will be used. The linear-phase requirement is essential since the temporal properties of the waveforms must be preserved in order that a meaningful correlation operation be obtained. The practical implementation of the optimum classifier against white noise is shown in Fig. VI-2. The detailed characteristics of the linear-phase filters are provided in the Conclusions of this section on p. 73.

Implicit in the realization illustrated in Fig. VI-2 is the estimation of the pitch period of a voiced waveform so that the additional discrimination inherent in the comb filter can be exploited. A further simplification in processor complexity can be obtained simply by omitting the comb

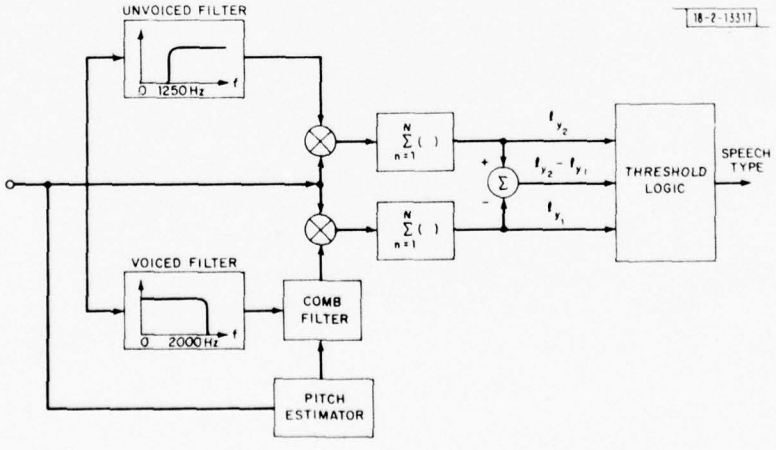


Fig. VI-2. Practical realization of optimum speech classifier.

filter and relying on the spectral orthogonality of the two speech types. However, since the periodicity of the voiced-speech process is a potentially powerful classification discriminant, for theoretical completeness, it is worthwhile to develop a practical algorithm to exploit it. Since this necessitates an estimate of the pitch period, a brief exposition of an optimum pitch-estimation algorithm will be presented.

E. PITCH ESTIMATION

Voiced speech was modeled as a periodic random process in the sense that $R_v(k) = R_v(k + T)$ for some pitch period T. This means that almost every sample function in the ensemble is periodic with period T. Therefore, the voiced speech signal v(n) can be modeled as

$$v(n) = q(n)_{\text{mod}T} \tag{VI-20}$$

where q(1), q(2), . . . , q(T) are completely unknown. Of course, to be faithful to the random process formulation of voiced speech, the quantities q(k) should be treated as correlated random variables. However, to keep the estimation problem mathematically tractable, the correlation properties will be ignored at first. The voiced-speech data are therefore taken to be

$$y(n) = v(n) + w(n) \tag{VI-21}$$

where w(n) represents white Gaussian noise, and v(n) is given by Eq. (VI-20). Based on N samples of these data, the parameters q(1), q(2), . . . , q(T) and T are to be estimated.

The above formulation of the pitch-estimation problem was formulated and solved by Wise, Caprio, and Parks.[10] Using the maximum-likelihood estimation rule they minimized the cost function

$$\begin{aligned} D(\underline{q}, T) &= \sum_{n=1}^{N} [y(n) - v(n)]^2 \\ &= \sum_{n=1}^{N} y^2(n) - 2 \sum_{k=1}^{T} \sum_{m=0}^{M-1} y(k + mT)\, v(k + mT) \\ &\quad + \sum_{k=1}^{T} \sum_{m=0}^{M-1} v^2(k + mT) \quad . \end{aligned} \tag{VI-22}$$

In order to simplify the derivation, it has been assumed that N = MT, M an integer.† From the periodicity condition $v(k + mT) = q(k)_{\text{mod}T}$, then Eq. (VI-22) reduces to

$$D(\underline{q}, T) = \sum_{n=1}^{N} y^2(n) - 2 \sum_{k=1}^{T} q(k) \sum_{m=0}^{M-1} y(k + mT) + M \sum_{k=1}^{T} q^2(k) \quad . \tag{VI-23}$$

† The more general case is tedious, and contributes little to the final result.

Since the basic voiced-speech waveform q(1), . . . , q(T) has been assumed completely unknown (i.e., the correlation properties have been ignored†), then, for the fixed T, the minimizing values are obviously

$$\hat{q}(k) = \frac{1}{M} \sum_{m=0}^{M-1} y(k + mT) \quad . \tag{VI-24}$$

The estimate of the voiced-speech waveform is therefore

$$\hat{v}(n|N) = \hat{q}(k)_{\text{mod}T} \tag{VI-25}$$

where the notation $\hat{v}(n|N)$ is used to denote the fact that all N measurements y(1), y(2), . . . , y(N) are used in developing the estimate of the voiced-speech waveform v(n), $n \leqslant N$. In that sense, the estimator is unrealizable.‡ The corresponding minimum value of the likelihood function is

$$D(T) = \sum_{n=1}^{N} [y(n) - \hat{v}(n|N)]^2 \tag{VI-26a}$$

$$= \sum_{n=1}^{N} y^2(n) - \sum_{n=1}^{N} \hat{v}^2(n|N) \quad . \tag{VI-26b}$$

Since $\hat{v}(n|N)$ can be interpreted as the output of a comb filter tuned to pitch period T when y(n) is the input, then the second term in Eq. (VI-26b) simply represents the energy at the output of this comb filter. Therefore, the optimum estimate of the pitch period can be obtained by constructing a bank of comb filters each tuned to a slightly different pitch period, and choosing as the estimate the pitch corresponding to the comb filter for which the output energy is largest.

It is important to keep in mind the fact that voiced-speech signals are at best quasi-periodic; hence, there is a definite limitation on the number of periods over which the averaging process is a meaningful operation. Since values of the pitch frequency generally fall within the range 70 to 300 Hz corresponding to pitch periods 3 to 15 msec long, and since the time required for a significant alteration in the vocal tract is approximately 20 msec, there can be 1 to 7 repetitions of the voiced-speech waveform. Therefore, the number of periods over which the data are averaged is a design parameter that must be chosen to carefully trade off the estimation accuracy and the quasi-periodic nature of the voiced-speech waveform.

A particularly important practical case corresponds to the assumption that the voiced-speech waveform is periodic for two successive periods. In this case, from Eqs. (VI-24) and (VI-25) the maximum-likelihood estimate of the voiced-speech signal is

$$\hat{v}(n|N) = \frac{1}{2} [y(n) + y(n - T)] \tag{VI-27}$$

---

† The more general case is treated by McAulay.[11]

‡ A realizable estimator that uses only the data up to time n is

$$\hat{v}(n|n) = \frac{1}{M} \sum_{m=0}^{M-1} y(n - MT) \quad .$$

which from Eq. (VI-26) results in the residual error

$$D(T) = \sum_{n=1}^{N} [y(n) - \hat{v}(n|N)]^2 = \frac{1}{4} \sum_{n=1}^{N} [y(n) - y(n-T)]^2 \quad . \tag{VI-28}$$

The estimate of the pitch period is then the value of T that minimizes D(T). This criterion has already been proposed for pitch estimation by Moorer[12] and by Ross et al.,[13] except that the squared difference has been approximated by the absolute magnitude difference function in order to achieve greater dynamic range and computational speed. Experimental results have shown that the quality of the pitch estimates is roughly equivalent to that of the cepstral method, and successful operation has also been demonstrated in strong noise environments. For this reason, it is conjectured that Eqs. (VI-24) through (VI-26) represent a possible solution to the problem of robust pitch estimation. To see this, suppose that the true pitch period is T_o; then, the observed data are

$$y(n) = v(n;T_o) + w(n) \tag{VI-29}$$

where $v(n;T_o) = q(k)_{\text{mod}T_o}$. The output of the comb filter tuned to pitch period T is

$$\hat{v}(n;T) = \frac{1}{M} \sum_{m=0}^{M-1} v(n-mT;T_o) + \frac{1}{M} \sum_{m=0}^{M-1} w(n-mT) \quad . \tag{VI-30}$$

The noise signal at the output of the comb filter is

$$\eta(n;T) = \frac{1}{M} \sum_{m=0}^{M-1} w(n-mT) \quad . \tag{VI-31}$$

As long as the correlation time of the noise process is less than the minimum pitch period of interest, then, if w(n) has variance σ^2, $\eta(n;T)$ will have variance σ^2/M. For the comb filter tuned to pitch T_o, the output signal is

$$\hat{v}(n;T_o) = q(k)_{\text{mod}T_o} + \eta(n;T_o) \quad . \tag{VI-32}$$

Therefore, there is an M:1 increase in SNR as a result of using the comb filter. Applied to the two-pulse canceler in Eq. (VI-29) (i.e., the AMDF), a 3-dB improvement in SNR is obtained for the class of noise processes whose correlation times are less than the minimum pitch period of interest.

Although originally proposed as a pitch estimation criterion based on ad hoc considerations, the maximum-likelihood theory shows that the average squared difference function is optimum and robust when the voiced-speech waveform is modeled as a deterministic quasi-periodic waveform with periodicity extending over two periods. The major limitation in using the two-pulse comb filter (i.e., the AMDF) is the not-infrequent occurrence of pitch doubling which occurs when the voiced speech is periodic for at least four pitch periods. At the expense of increasing the length of the speech buffer, an M-pulse comb filter, $M \geq 3$, can be used to reduce the rate at which pitch doubling errors occur.

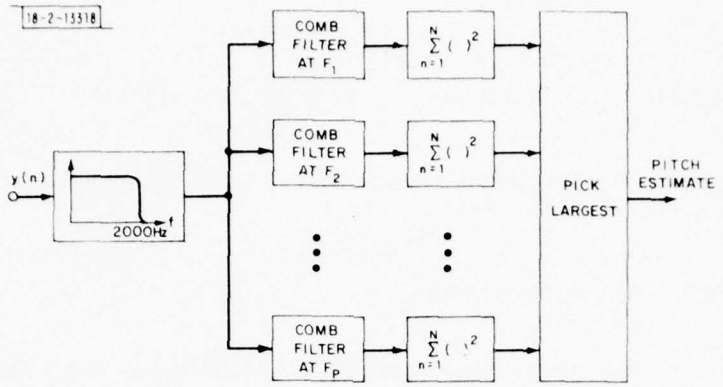


Fig. VI-3. Practical implementation of optimum pitch estimator.

A further enhancement in the pitch estimate can be obtained by using the low-pass voiced-speech filter to increase the pitch estimator SNR. This corresponds to exploitation of the global correlation properties of voiced speech. The approximate matched filter configuration of the pitch detector is shown in Fig. VI-3.

F. THE OPTIMUM CLASSIFIER AGAINST COLORED NOISE

There are several examples in which speech in nonwhite acoustic background noise can be effectively classified using the algorithm that was defined to be optimum against white noise. In particular, whenever the SNR is high, the white-noise classifier will yield acceptable performance. There are some cases, particularly if the SNR is low and the noise is highly correlated, where significant improvements can be achieved by taking the spectral characteristics of the noise into account. In this section, the structure of the optimum classifier will be derived for the colored-noise case and then reasonable practical approximations will be deduced in order to simplify the complexity of the signal processor.

For this classification problem, the data corresponding to hypothesis H_i are

$$y(n) = s_i(n) + w_c(n) + w(n) \qquad i = 1, 2, 3 \tag{VI-33}$$

where $w_c(n)$ denotes the colored noise present on all three hypotheses. Note that a white-noise component w(n) is also incorporated into the model to avoid mathematical problems relating to singular solutions. The standard approach to this problem is to precede all the processing by a whitening filter and then apply the white-noise solution. This was the approach taken by McAulay,[11] which, although mathematically correct, encounters practical difficulties because the whitening filter essentially pre-emphasizes the speech data. As has already been discussed, this can cause the higher formants of voiced speech to acquire the same attributes as unvoiced speech, which makes classification difficult. McAulay and Yates[14] derived an estimator-correlator classifier that does not require a whitening prefilter. Drawing on their results and those developed in Sec. C above, two sufficient statistics are computed, namely:

$$\ell_{zi} = \sum_{n=1}^{N} z(n)\, \hat{s}_i(n) \qquad i = 2, 3 \tag{VI-34}$$

where

$$\hat{s}_i(n) = \sum_{k=-\infty}^{\infty} h_i(n-k)\, y(k) \qquad i = 2, 3 \tag{VI-35}$$

is the linear least-squared error unrealizable estimate of $s_i(n)$ based on the data $y(n) = s_i(n) + w_c(n) + w(n)$, and where

$$z(n) = \sum_{k=-\infty}^{\infty} h_c(n-k)\, y(k) \tag{VI-36}$$

is the result of passing y(n) through the clutter-rejection filter $h_c(n)$. It has been implicitly assumed that the speech and noise processes are independent and quasi-stationary. The transfer functions of the filters are[14]

$$H_i(f) = \frac{G_i(f)}{G_i(f) + G_c(f) + N_o/2} \qquad i = 2, 3 \tag{VI-37}$$

$$H_c(f) = 1 - \frac{G_c(f)}{G_c(f) + N_o/2} = \frac{N_o/2}{G_c(f) + N_o/2} \tag{VI-38}$$

where $G_c(f)$, $G_2(f)$, and $G_3(f)$ represent the power spectra for the colored-noise, unvoiced-speech, and voiced-speech processes, respectively. The second term in Eq. (VI-38) is precisely the linear least-squares unrealizable estimator of $w_c(n)$ based on the signal $w_c(n) + w(n)$. Therefore, the clutter filter attempts to remove the colored noise from the data before performing the correlation operation. The optimum classifier structure is shown in Fig. VI-4. The classification rule is similar to that derived for white noise, Eq. (VI-9), except that the sufficient statistics are now ℓ_{z_2} and ℓ_{z_3} instead of ℓ_{y_2} and ℓ_{y_3}.

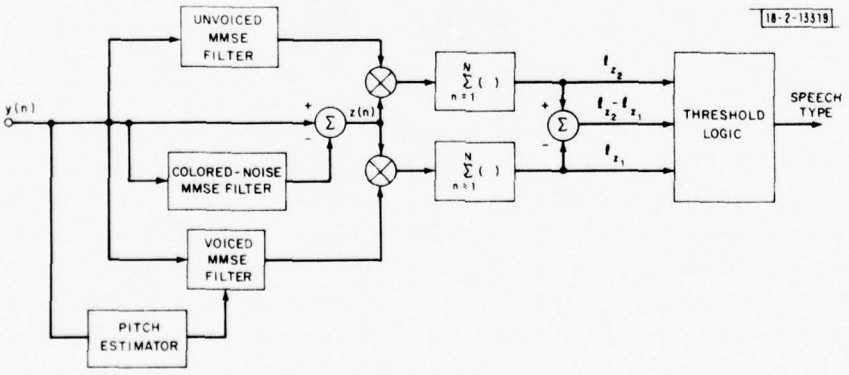


Fig. VI-4. Optimum speech classifier against colored noise.

G. PRACTICAL IMPLEMENTATION OF THE ESTIMATOR-CORRELATOR SPEECH CLASSIFIER

The arguments for simplifying the processing of voiced and unvoiced speech proceed along the same lines as those made for the white-noise case. In particular, if knowledge of pitch is

available, the spectral harmonics of voiced speech are matched by using a comb filter in cascade with the Wiener filter designed on the basis of the spectral envelope. Therefore, the voiced-speech estimator filter is

$$H_v(f) = C(f;\hat{T}) \frac{G_v(f)}{G_v(f) + G_c(f) + N_o/2} \tag{VI-39}$$

where $C(f;\hat{T})$ is the comb filter tuned to the most-recent pitch estimate $\hat{T}$. For unvoiced speech, the estimator filter is†

$$H_u(f) = \frac{G_u(f)}{G_u(f) + G_c(f)} \quad . \tag{VI-40}$$

Lacking knowledge of the exact form of $G_v(f)$ and $G_u(f)$, a good first approximation is to use the linear-phase low-pass (cutoff above 2000 Hz) and high-pass (cutoff below 1250 Hz) filters in the voiced- and unvoiced-speech channels as was done in the white-noise case. This insures the spectral orthogonality of the two speech channels and enhances the speech-to-noise ratio whenever the noise spectrum lies outside the filter passbands. For colored noise, however, it is possible that all the noise energy will lie within the filter passbands, in which case no speech enhancement will occur if only the fixed filters are used. Somehow, additional processing tuned to reject the clutter will have to precede the fixed filters in the speech channels. To develop a clue as to the form of the clutter processor, it is necessary to re-examine Eqs. (VI-39) and (VI-40). Letting $G_2(f) \approx G_u(f)$ and $G_3(f) \approx G_v(f)$, then the unvoiced- and voiced-speech Wiener filters can be written as

$$H_i(f) = \frac{G_i(f)}{G_i(f) + G_c(f)} \quad . \tag{VI-41}$$

Realization of these filters requires that the speech and noise spectra be known. Since the noise statistics can be measured during the silent intervals, it is reasonable to assume that the clutter spectrum is known. Unfortunately, a priori estimates of the speech spectra are not available unless long-term averages are determined from training sets. When detailed knowledge of the frequency distribution of the speech is unavailable, a conservative approach is to model the speech as white noise thereby having a flat spectrum. Letting

$$G_i(f) = \alpha_i \qquad i = 2, 3 \tag{VI-42}$$

and substituting this into Eq. (VI-41) results in the filters

$$H_i(f) = \frac{\alpha_i}{G_c(f) + \alpha_i} \qquad i = 2, 3 \quad . \tag{VI-43}$$

Since $H_i(f) \approx 0$ whenever $G_c(f) \gg \alpha_i$, and $H_i(f) \approx 1$ whenever $G_c(f) \ll \alpha_i$, Eq. (VI-43) can be interpreted as a notch filter tuned to reject "most" of the clutter energy. When the speech-to-noise ratio is large, little clutter rejection is needed and α_i should be large, since this results in a passband filter. When the speech-to-noise ratio is small, then the clutter must be rejected whatever the cost in speech distortion, which necessitates a small value for α_i. It follows, therefore, that the parameter α_i should be proportional to the speech-to-noise ratio. Since the

† The effects of the artificial white-noise term have been neglected at this point, since there is no problem with singular solutions.

clutter power is known from the silent intervals, estimates of the speech-to-noise ratio can be made from the data frame being analyzed. In this mode, the distinction between voiced and unvoiced speech disappears and only a single parameter value and clutter filter need be determined. In this sense, the clutter filter represents an adaptive prefilter whose output, in a conservative sense, represents the best available estimate of the speech waveform.

The results of this discussion are summarized in Fig. VI-5 which shows the practical realization of the optimum classifier operating against a colored-noise background. Except for the clutter filters in the reference and speech channels, the processing is identical to that used in the white-noise case. Since selection of the tuning parameters α_c and α_s depends on the noise statistics, further discussion regarding their selection will be deferred to Sec. H below.

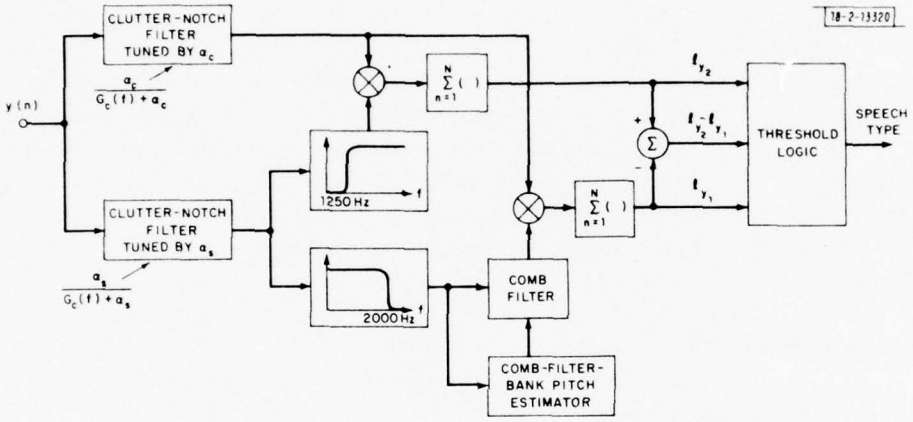


Fig. VI-5. Practical realization of optimum speech classifier against colored noise.

The only problem that remains to be discussed is the calculation of the clutter-filter impulse response from Eq. (VI-43). The most straightforward approach is to solve the Wiener-Hopf equation

$$\sum_{k=-\infty}^{\infty} a_k \left[R_c(k-j) + \alpha\,\delta(k-j)\right] = \alpha\,\delta(j) \qquad -\infty < j < \infty \quad . \tag{VI-44}$$

If the impulse response is truncated at $\pm p$, the $2p+1$ coefficients a_k can be found by solving Eq. (VI-44) numerically using the Levinson Recursion. Another approach is to fit an all-pole spectrum to $G_c(f) + \alpha$ using Linear Prediction techniques and use the spectral coefficients to determine the clutter filter. For this method, the LPC spectral estimate of $G_c(f) + \alpha$ can be obtained by solving

$$\sum_{k=1}^{p} a_k \left[R_c(k-j) + \alpha\,\delta(k-j)\right] = R_c(j) \qquad 1 \leqslant j \leqslant p \quad . \tag{VI-45}$$

This equation can be solved efficiently using the Levinson Recursion and results in a p-pole fit to the clutter spectrum. The estimated spectrum is

$$\widehat{G_c(z) + \alpha} = \frac{\sigma}{A(z)\, A^*(z)} \tag{VI-46}$$

where

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \tag{VI-47}$$

which corresponds to the Inverse Filter in the usual LPC analysis. Substituting Eq. (VI-46) into Eq. (VI-43) results in the Wiener filter

$$H(z) = \frac{\alpha}{\sigma}\, A(z)\, A^*(z) \quad . \tag{VI-48}$$

Letting y(n) denote the input sequence and $\hat{s}(n)$ the output sequence, then

$$\begin{aligned} \hat{S}(z) &= \frac{\alpha}{\sigma}\, A(z)\, A^*(z)\, Y(z) \\ &= \frac{\alpha}{\sigma}\, A(z)\, X(z) \end{aligned} \tag{VI-49}$$

where

$$X(z) = A^*(z)\, Y(z) \quad . \tag{VI-50}$$

Since the LPC coefficients $\{a_k\}$ are real

$$A^*(z) = 1 - \sum_{k=1}^{p} a_k z^{k} \tag{VI-51}$$

and

$$x(n) = y(n) - \sum_{k=1}^{p} a_k y(n + k) \tag{VI-52}$$

$$\hat{s}(n) = \frac{\alpha}{\sigma}\, [x(n) - \sum_{k=1}^{p} a_k x(n - k)] \quad . \tag{VI-53}$$

Therefore, the unrealizable Wiener filter can be implemented by the cascade combination of an inverse filter that operates on p samples of future data and an inverse filter that operates on p samples of past data. A p-sample buffer must therefore be available to provide for the future data. The advantage of this approach is that the length of the impulse response is completely determined on the basis of the number of poles required to fit the clutter spectrum.

H. EXPERIMENTAL RESULTS

The signal-processing concepts developed in the previous sections were evaluated experimentally using speech data that were corrupted by Airborne Command Post (ACP) noise. Not only does this provide a good pedagogical tool for illustrating the filtering ideas, but it represents

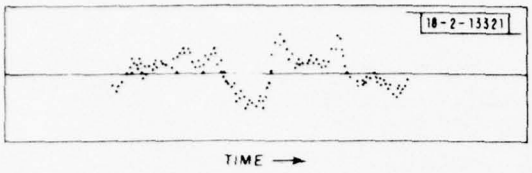


Fig. VI-6(a). ACP noise sample function.

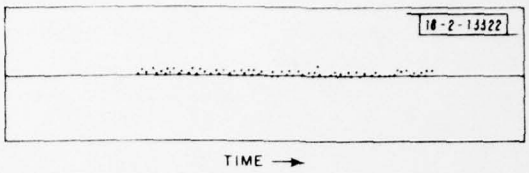


Fig. VI-6(b). Reference channel output.

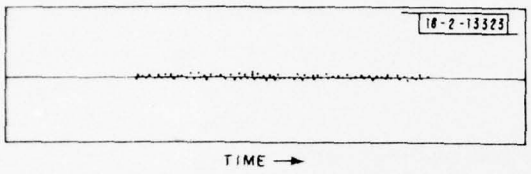


Fig. VI-6(c). Unvoiced-speech channel output.

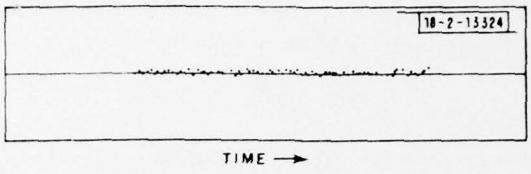


Fig. VI-6(d). Voiced-speech channel output.

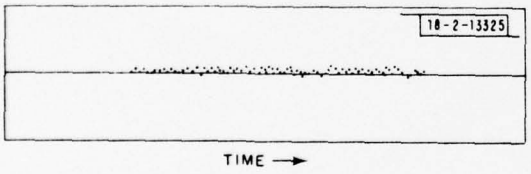


Fig. VI-6(e). Prefilter output.

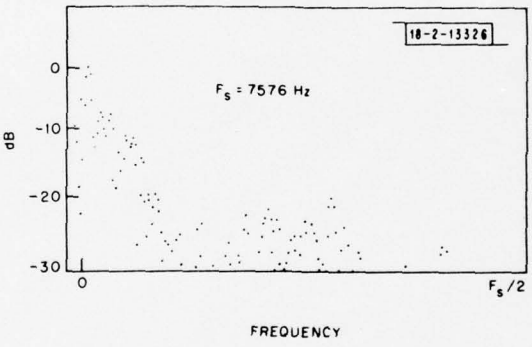


Fig. VI-6(f). ACP noise power spectrum.

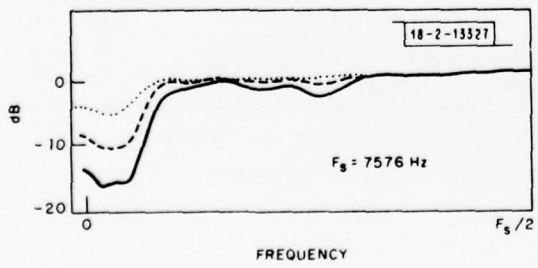


Fig. VI-6(g). Clutter-filter frequency response.

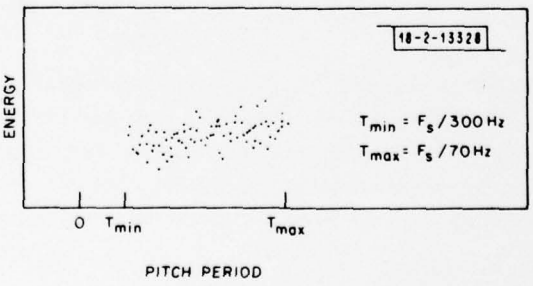


Fig. VI-6(h). Comb-filter response.

an important real-world speech-encoding environment which is not adequately solved using state-of-the-art vocoder technology.

The noisy-speech data were sampled every 132 μsec (7575 Hz), and 158 samples were collected to define a 20-msec frame. Figure VI-6(a) illustrates a 20-msec sample function of ACP noise; Fig. VI-6(f) is a plot of the magnitude of its Fourier transform measured in decibels. The correlation function of the m^{th} frame (i.e., the current frame) of noise data was computed from

$$R_y(k;m) = \sum_{n=0}^{N-1-k} x(n)\, x(n+k) \qquad k = 0, 1, \ldots, p \quad ; \quad m = 1, 2, \ldots \tag{VI-54}$$

where $x(n)$ is the Hamming weighted version of the input data $y(n)$. A first-order smoothed correlation function was then computed from

$$\bar{R}_c(k;m) = \frac{1-\gamma}{1-\gamma^m}\left[R_y(k;m) + \gamma \bar{R}_c(k;m-1)\right] \quad . \tag{VI-55}$$

In general, the weighting constant γ should be chosen to reflect the quasi-stationarity of the noise random process. For ACP noise, $\gamma = 0.95$ was chosen arbitrarily and seemed to produce good results.

From Eq. (VI-38), the clutter filter in the reference channel was given by

$$H_c(z;m) = \frac{\alpha_c(m)}{G_c(z) + \alpha_c(m)} \quad . \tag{VI-56}$$

The impulse response was found using Linear Prediction techniques as described in the previous section. This necessitates solving the Wiener-Hopf prediction equation

$$\sum_{k=1}^{p} a_k \left[\bar{R}_c(k-j;m) + \alpha_c(m)\, \delta(k-j)\right] = \bar{R}_c(j;m) \qquad 1 \leqslant j \leqslant p \tag{VI-57}$$

using the long-term averaged correlation function computed at the last frame (i.e., the m^{th} frame). A whole class of clutter filters can be obtained simply by varying the parameter $\alpha_c(m)$. Typical transfer functions from this class are shown in Fig. VI-6(g) for three values of α_c. It was found that the clutter filter defined for the value $\alpha_c(m) = \bar{R}_c(0;m)$ worked well for ACP noise. For other noise types, other values would probably be more appropriate. A little experimentation is therefore required to tune the clutter filter to different noise processes.

The unvoiced- and voiced-speech channels are preceded by another clutter-rejection filter given by Eq. (VI-43), namely,

$$H_s(z;m) = \frac{\alpha_s(m)}{G_c(z) + \alpha_s(m)} \tag{VI-58}$$

where α_s is chosen to be proportional to the speech-to-noise ratio measured for the current frame of data (i.e., the m^{th} frame). Since $R_y(0;m)$ represents a measure of the speech-plus-noise energy for the current frame of data, and since $\bar{R}_c(0;m)$ represents a measure of the long-term averaged noise energy, then a reasonable estimate for the speech-to-noise energy is

$$\hat{\xi}(m) = R_y(0;m) - \bar{R}_c(0;m) \quad . \tag{VI-59}$$

It is possible that the energy in any one 20-msec sample function will be less than the average clutter energy, especially if that sample function contains noise alone or noise plus unvoiced speech. Therefore, provision must be made to bound the clutter-notch parameter α_s away from zero. A reasonable scheme is to pick

$$\alpha_s(m) = \max [\hat{\xi}(m), \alpha_c(m)] \tag{VI-60}$$

which guarantees that the speech-clutter-filter notch will never be deeper than that in the reference channel. As before, the impulse response was found using the Linear Prediction power spectrum which was obtained by solving the Wiener-Hopf predictor equation (VI-57) using α_s instead of α_c.

The output of the speech clutter filter was then used as the input to the high- and low-pass filters characterizing the unvoiced- and voiced-speech processing channels, respectively. The filters were both 21-tap linear-phase digital filters designed using the Parks-McClellan algorithm.[15] The impulse responses and frequency characteristics are specified in the Conclusions of this section on p. 73. No attempt was made to optimize the filter design. The outputs of the reference-channel clutter filter z(n) and the unvoiced- and voiced-speech filters $\hat{u}(n)$, $\hat{v}(n)$ are shown in Figs. VI-6(b) through (d). According to Eq. (VI-34), the outputs of the speech filters were then correlated with the output of the reference-channel clutter filter to form the detection statistics:

$$\ell_1(m) = \sum_{n=1}^{N} z(n)\, \hat{u}(n) \tag{VI-61a}$$

$$\ell_2(m) = \sum_{n=1}^{N} z(n)\, \hat{v}(n) \tag{VI-61b}$$

$$\ell_3(m) = \ell_u(m) - \ell_v(m) \quad . \tag{VI-61c}$$

It should be noted that the comb filter has been left out of the voiced-speech processing channel. This decision was made to show that good classifier performance could be obtained without having to make a pitch estimate which simplifies the classifier processing, which is necessary for some applications.

The detection thresholds were obtained by driving the system with ACP noise for 15 data frames (0.3 sec). This is the only training cycle required by the processor, and should be relatively easy to meet in practice because there is always a speech-free interval before a talker actually speaks into the encoding device after having turned the machine on. Averaged detection statistics for the training noise are computed from

$$\overline{\ell}_i(m) = \frac{1-\gamma}{1-\gamma^m} [\ell_i(m) + \gamma \overline{\ell}_i(m-1)] \qquad i = 1, 2, 3 \tag{VI-62}$$

with $\gamma = 0.95$ as before. The detection thresholds were then chosen to be

$$\lambda_i(m) = 1.5\overline{\ell}_i(m) \qquad i = 1, 2$$

$$\lambda_3(m) = \overline{\ell}_1(m) - \overline{\ell}_2(m) \tag{VI-63}$$

which allows for moderate statistical fluctuations. After the first 15 data frames of noise have been processed (m = 15) and the initial threshold setting computed, the classification process is initiated. The next frame of data is processed and the detection statistics $\ell_i(m+1)$ are computed. If $\ell_1(m+1) < \lambda_1(m)$, and $\ell_2(m+1) < \lambda_2(m)$, then the data are classified as silence and the clutter correlation function (VI-55) and the detection thresholds (VI-62) and (VI-63) are updated. If $\ell_1(m+1) > \lambda_1(m)$, or $\ell_2(m+1) > \lambda_2(m)$, then speech is declared present and neither the clutter correlation function nor the detection thresholds is changed. No updating is done until the next frame of silence is detected. This procedure allows the classifier to track noise processes whose statistics vary slowly with time. Such a classifier structure is often referred to as a decision-directed detector since it tells itself when to alter its structure. It becomes evident, therefore, that the detection thresholds should be set low even at the expense of a high false-alarm rate (declaring noise as speech is a false alarm). It would be a more serious error if the classifier declared speech as noise since, then, all the clutter filters and detection thresholds would be tuned to reject speech. Fortunately, this malign event rarely occurred for ACP noise, and when it did the noise always completely overpowered the speech so that little change in the filter structures occurred.

The effects of the three filtering channels on the three speech types will be examined for some typical cases to develop a feeling for the classifier operation. Figure VI-6(a) is a plot of a 20-msec-input sample function of ACP noise; Fig. VI-6(f) is the corresponding short-term power spectrum. Figure VI-6(g) is a plot of the adaptive clutter-filter transfer function in the reference channel (the adaptive prefilter). For ACP noise input, it has adapted in such a way as to make a -10-dB null at the clutter frequencies. Figures VI-6(b) through (d) show the respective outputs of the reference channel, the high-pass filtered unvoiced-speech channel, and the low-pass filtered voiced-speech channel.

As was described in the previous section, the output of the speech-channel clutter filter represents a minimum mean-squared-error estimate of the input speech. Figure VI-6(e) shows a plot of the prefilter output in response to ACP noise at the input. Of course, with high probability the classifier will classify the frame as silence; hence, one has the option of setting the prefilter output to zero, which removes the residual noise completely.

Although the comb-filter discriminator was not used in the classifier, it remains of interest to evaluate the robustness of the maximum-likelihood pitch estimator in ACP noise. This was done by applying the output of the low-pass filter $\hat{v}(n)$ to a bank of two-pulse comb filters covering the range from 70 to 300 Hz. Figure VI-6(h) is a plot of the energy at the output of the comb filters as a function of the pitch period for the ACP noise sample.

The same sequence of data is plotted in Figs. VI-7(a) through (h) and VI-8(a) through (h) for 20-msec frames of unvoiced and voiced speech, respectively. Figures VI-7(a) and (f) show that the unvoiced-speech-to-noise ratio is less then 0 dB (it is roughly -3 dB), yet Fig. VI-7(e) shows that the prefilter has removed a significant portion of the clutter waveform while allowing the unvoiced-speech waveform to pass relatively undisturbed. Figures VI-8(a) and (f) show that the voiced-speech-to-noise ratio is quite large (it is roughly 9 dB). Figure VI-8(g) shows that the prefilter transfer function is adjusted to allow most of the speech to pass, even though its spectrum overlaps that of the ACP noise. This shows the advantage of the adaptive prefilter. Had a fixed clutter filter been used, the voiced-speech waveform would have been distorted unnecessarily. Figure VI-8(h) shows that the pitch estimate is perturbed very little by the presence of ACP noise. In general, the only significant pitch errors found were the effects of pitch

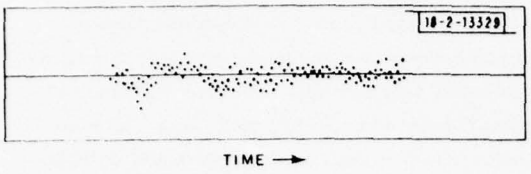


Fig. VI-7(a). Unvoiced-speech sample function.

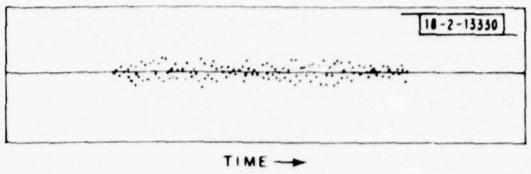


Fig. VI-7(b). Reference channel output.

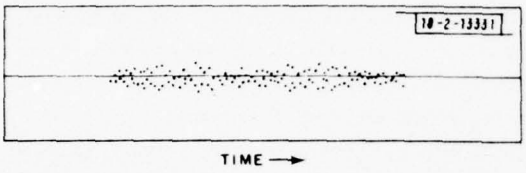


Fig. VI-7(c). Unvoiced-speech channel output.

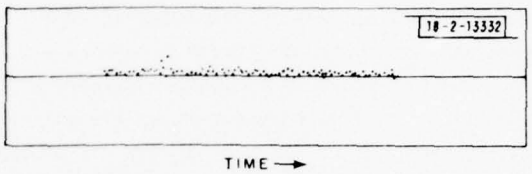


Fig. VI-7(d). Voiced-speech channel output.

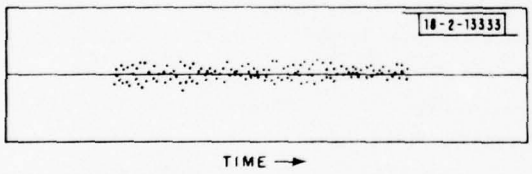


Fig. VI-7(e). Prefilter output.

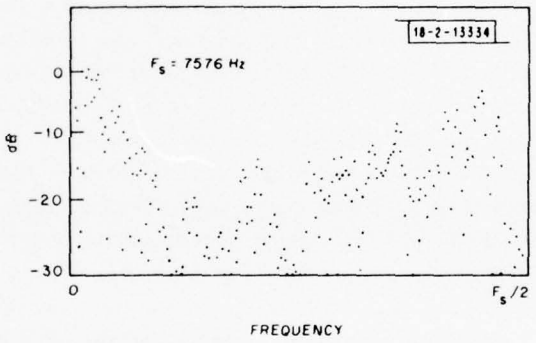


Fig. VI-7(f). Unvoiced-speech power spectrum.

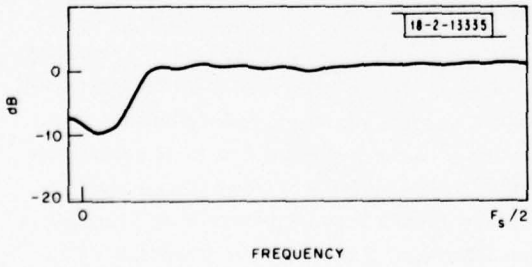


Fig. VI-7(g). Prefilter frequency response.

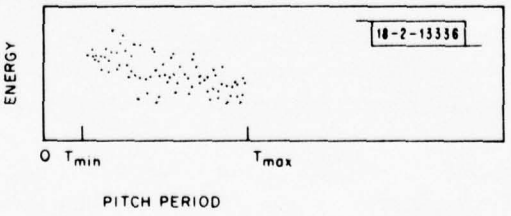


Fig. VI-7(h). Comb-filter response.

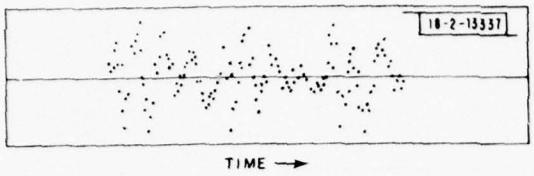


Fig. VI-8(a). Voiced-speech sample function.

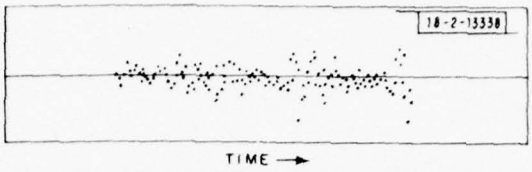


Fig. VI-8(b). Reference channel output.

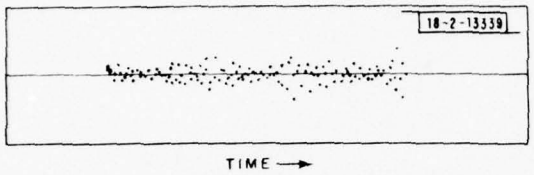


Fig. VI-8(c). Unvoiced-speech channel output.

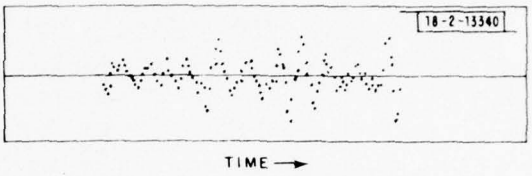


Fig. VI-8(d). Voiced-speech channel output.

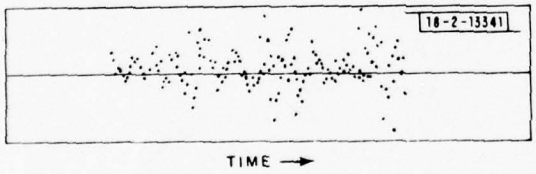


Fig. VI-8(e). Prefilter output.

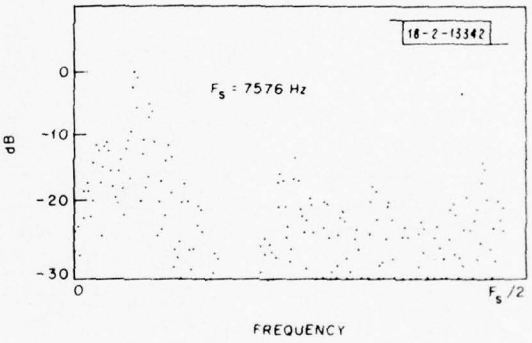


Fig. VI-8(f). Voiced-speech power spectrum.

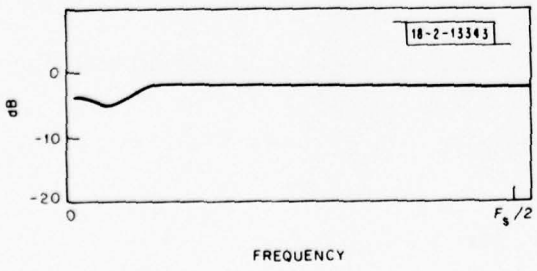


Fig. VI-8(g). Prefilter frequency response.

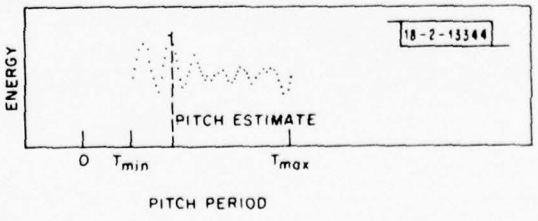


Fig. VI-8(h). Comb-filter response.

TABLE VI-1
CLASSIFIER PERFORMANCE STATISTICS

True \ Estimated	Silence	Unvoiced	Voiced
Silence	405	14	24
Unvoiced	4	43	2
Voiced	1	5	170
Unvoiced-Voiced	0	0	6

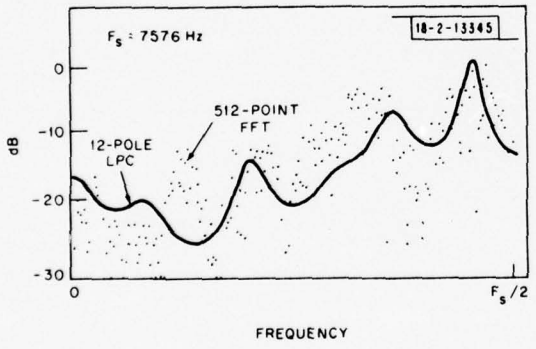


Fig. VI-9. Unvoiced-speech prefilter output spectra.

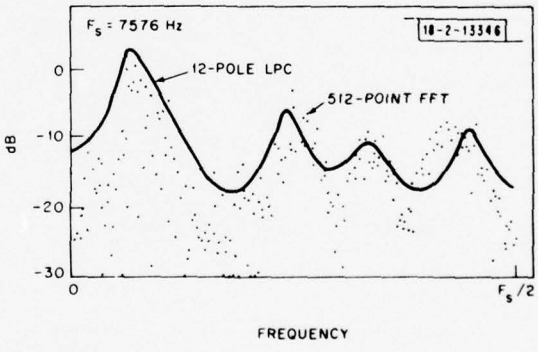


Fig. VI-10. Voiced-speech prefilter output spectra.

doubling which occurred intermittently near the ends of a voiced sound. Figure VI-8(e) shows how the prefilter attempts to reproduce the voiced-speech waveform.

Having established the basic characteristics of the classifier, our next step is to evaluate the frame-to-frame performance when an ACP noise-corrupted utterance is applied to the input. Classification errors were obtained by determining the true speech type by visually examining the waveform, power spectrum, and comb-filter energy contour for each 20-msec sample function. Statistics were accumulated for a total of three utterances spoken by three male speakers in different ACP noise environments. The results, from which the false-alarm probability (declare speech given silence) is estimated to be 9.4 percent, are tabulated in Table VI-1. The miss probability (declare silence given speech) is 2.3 percent. The misses mainly occurred for unvoiced speech that had been completely overpowered by the noise (~ -10-dB speech-to-noise ratio). Erroneous classifications (voiced $\leftrightarrow$ unvoiced) occurred at the rate of 3 percent. Whenever a frame represented a mixture of voiced and unvoiced speech, the classifier always chose in favor of voiced speech. This event could be reduced significantly by reducing the frame period (10 vs 20 msec). Although these statistics have been gathered for a relatively small ensemble, the general impression is that the performance is quite good.

Another aspect of the experimental program was the recovery and synthesis of noise-corrupted speech using Linear Prediction techniques. The voiced-unvoiced decisions and the pitch estimates were derived using the methods described here. The LPC filter coefficients were estimated from the prefilter output waveform. For the case of noise-corrupted unvoiced speech, Fig. VI-7(a) for example, the prefilter output is shown in Fig. VI-7(e). Its short-term power spectrum is shown in Fig. VI-9 which, when compared with that for the input unvoiced speech plus ACP noise – Fig. VI-7(f), clearly demonstrates the action of the adaptive prefilter in eliminating the clutter. The LPC power-spectrum estimate is also plotted in Fig. VI-9 and shows that the synthetic speech is likely to reproduce the original unvoiced speech. Of course, the ACP noise will cause the spectral estimate to be somewhat distorted, but the perception of the additive ACP noise will have disappeared. It is for this reason that the synthetic speech is perceived to be "noise-free."

Similar results are obtained for the voiced-speech sample function shown in Fig. VI-8(a). The short-term power spectrum of the prefilter output, Fig. VI-8(e), is plotted in Fig. VI-10 and should be compared with the voiced speech plus noise power spectrum shown in Fig. VI-8(f). The corresponding LPC spectrum shown in Fig. VI-10 shows the distortion in the first format due to the presence of the ACP noise.

LPC synthetic speech was generated for a number of utterances recorded in ACP noise. Compared with LPC speech in which no adaptive prefiltering was employed, an improvement in intelligibility was obtained.

I. CONCLUSIONS

Using statistical decision theory, a new speech-classification algorithm has been developed in the form of an estimator-correlator receiver. The structure is robust in the sense that it can adapt to time-varying noise fields in which the SNR can be quite low (less than 10 dB). For noiseless speech, the classifier simply involves two fixed filters and requires no pitch estimation or linear-prediction-analysis parameters. For noisy speech, clutter filters must be added to the speech and reference channels. The reference clutter filter is developed on the basis of an initial 0.3-sec sample of noise data, while the other adapts to the speech-plus-noise statistics

TABLE VI-2

VOICED- AND UNVOICED-FILTER IMPULSE RESPONSES

Impulse Response	Unvoiced Filter	Voiced Filter
h(1)	−0.21511067E-01	−0.38655568E-02
h(2)	0.55939741E-02	−0.32053679E-01
h(3)	0.21661893E-01	0.23418449E-01
h(4)	0.39310634E-01	0.13665602E-01
h(5)	0.45899481E-01	−0.42199165E-01
h(6)	0.29383000E-01	0.73566064E-02
h(7)	−0.15331455E-01	0.66053927E-01
h(8)	−0.82191288E-01	−0.65457523E-01
h(9)	−0.15448785E+00	−0.84543467E-01
h(10)	−0.21035391E+00	0.30347985E+00
h(11)	0.76869851E+00	0.59147525E+00

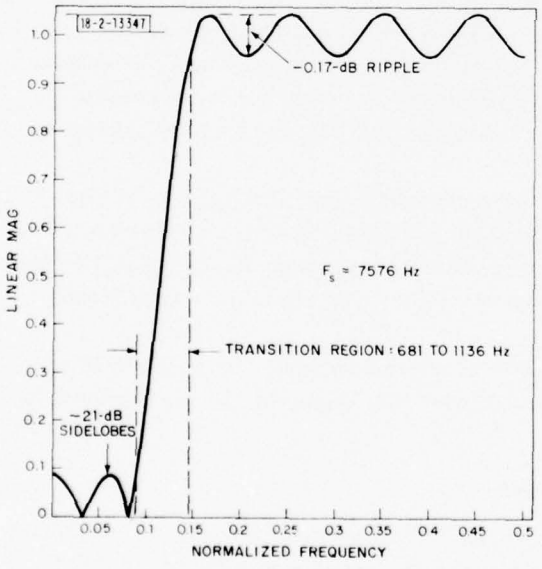


Fig. VI-11. Unvoiced high-pass filter.

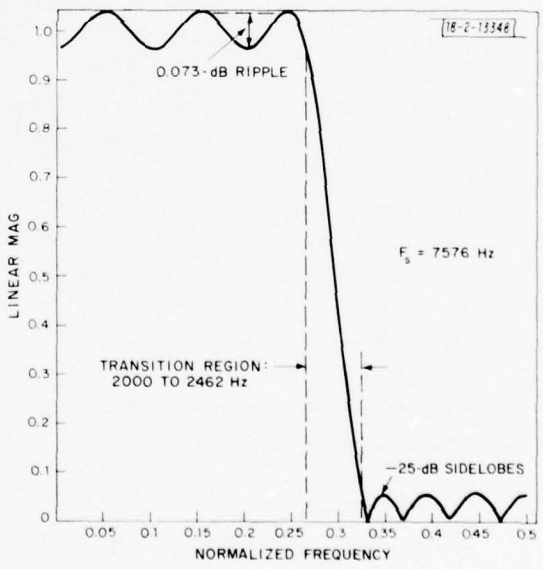


Fig. VI-12. Voiced low-pass filter.

calculated for each frame. If a frame is classified as noise, the reference-channel filter is updated so that time-varying noise statistics can be tracked.

The output of the speech-channel clutter filter represents an improved estimate of the input speech in the sense that much of the additive noise has been canceled from the signal. By applying Linear Prediction techniques to this waveform, more intelligible synthetic speech can be obtained.

A relatively thorough (non-real-time) evaluation of the classifier and adaptive prefilter was conducted for ACP noise, and surprisingly good results were obtained. Based on a limited number of listening tests, the LPC synthetic speech using the prefilter output was found to be more intelligible than the LPC synthesis of the original noisy speech.

No attempt was made to optimize the design of the fixed-voiced (low-pass) and unvoiced (high-pass) filters. The unvoiced- and voiced-speech Wiener filters were approximated by 21-tap linear-phase high- and low-pass filters designed using the Parks-McClellan algorithm. Impulse responses used in the experimental program are given in Table VI-2 [$h(n) = h(-n)$]. The magnitude of the frequency responses is shown in Figs. VI-11 and VI-12. A better approach would be to obtain long-term statistics for voiced and unvoiced speech and pick the filter length and passband edges to more closely represent the average spectral properties. Another useful study would be to investigate the possibility of using recursive filters with phase compensation to further simplify the processing.

Although a first-order attempt was made to improve the design of the clutter filters, other methods are undoubtedly possible. Additional insights are also needed in the selection of the clutter-filter design parameter; in this report, trial and error were used to make the selection.

Of course, the real test of any speech-processing algorithm is obtained in a real-time environment. This is the focus of the current effort.

REFERENCES

1. B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," J. Acoust. Soc. Am. 50, 637 (1971).
2. J. Makhoul, "Linear Prediction: A Tutorial Review," Proc. IEEE 63, 561 (1975).
3. J. D. Markel and A. H. Gray, Jr., "A Linear Prediction Vocoder Simulation Based Upon the Autocorrelation Method," IEEE Trans. Acoust., Speech, and Signal Processing ASSP-22, 124 (1974).
4. B. S. Atal and M. R. Schroeder, "Adaptive Predictive Coding of Speech Signals," Bell Syst. Tech. J. 49, 1973 (1970).
5. B. Gold, "Robust Speech Processing," Technical Note 1976-6, Lincoln Laboratory, M.I.T. (27 January 1976), DDC AD-A021899/0.
6. B. S. Atal and L. R. Rabiner, "A Pattern-Recognition Approach to Voiced-Unvoiced-Silence Classification with Applications to Speech Recognition," IEEE Trans. Acoust., Speech, and Signal Processing ASSP-24, 201 (1976).
7. H. L. Van Trees, Detection, Estimation and Modulation Theory, Part III (Wiley, New York, 1968).
8. Ibid., Part I (Wiley, New York, 1968).
9. N. Levinson, "The Wiener RMS Error Criterion in Filter Design and Prediction," J. Math. Phys. 25, 261 (1947).
10. J. D. Wise, J. R. Caprio, and T. W. Parks, "Maximum Likelihood Pitch Detection," Technical Report No. 7516, Department of Electrical Engineering, Rice University (17 October 1975).
11. R. J. McAulay, "Optimum Classification of Voiced Speech, Unvoiced Speech and Silence in the Presence of Noise and Interference," Technical Note 1976-7, Lincoln Laboratory, M.I.T. (3 June 1976), DDC AD-A028518/9.
12. J. A. Moorer, "The Optimum Comb Method of Pitch Period Analysis of Continuous Digitized Speech," IEEE Trans. Acoust., Speech, and Signal Processing ASSP-22, 330 (1974).
13. M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg, and H. J. Manley, "Average Magnitude Difference Function Pitch Extractor," IEEE Trans. Acoust., Speech, and Signal Processing ASSP-22, 353 (1974).
14. R. J. McAulay and R. D. Yates, "Realization of the Gauss-in-Gauss Detector Using Minimum-Mean-Squared-Error Filters," IEEE Trans. Inform. Theory IT-17, 207 (1971), DDC AD-729596.
15. J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters," IEEE Trans. Audio Electroacoust. AU-21, 506 (1973).

VII. TANDEMING AND IMPROVEMENT OF HIGH-RATE CODERS

Investigations of improved wideband speech digitizers (CVSD and APC) and improved tandem arrangements (CVSD-CVSD, LPC-CVSD) required the design of two distinct software systems on the Group 24 Univac-FDP facility.

For comparison of standard CVSD and modified CVSD algorithms, a design program was written for the FDP. This design program allowed two independent CVSD encoders to run on the same audio input samples with the parameters of one of the encoders easily modified. Both encoders drove a routine which computed an average mean-square error between the input and reconstructed speech waveforms. In this way, parameter changes in the CVSD algorithm could be evaluated by comparing the resulting mean-squared speech error with a "standard" CVSD algorithm error for a given audio input. The output reconstructed speech from both modified and unmodified algorithms could be played out of the FDP digital-to-analog (D/A) converter for real-time listening to the two outputs. Finally, the input waveform and both reconstructed waveforms could also be displayed for comparison and quantitative evaluation. Another program was written to deal with the wideband tandeming situation. This program could tandem up to four CVSD encoders with high-order elliptic low-pass filters between encoders to smooth intermediate waveforms and destroy synchrony between encoders. This was an accurate simulation of tandemed real-world CVSD units. In use, the first program described was used to converge to an "optimum" set of parameters (i.e., two time constants, and upper and lower slope changes), then the second program was used to evaluate the "optimum" encoder in a tandem environment.

CVSD parameter adjustment at 16 kbps was studied using this FDP-Univac facility. An audio tape was generated for several conditions of slope clamping and slope time constant as well as four times tandem. The following points can be made to summarize the study and the tape. First, there appear to be no "optimum" settings for the high- and low-slope clamps and the slope time constant, in the sense that large variations in these parameters (for example, 10:1 in the case of time constant) effect the output speech just perceptibly, and probably do not effect intelligibility at all. The upper clamp level interacts with the time constant to the degree that recovery to wide-dynamic-range changes is effected by a large time constant and a large upper clamp. Again, this is noticeable only with large changes in parameters. Probably the most noticeable effect of parameter changing occurs with the lower clamp, although intelligibility is degraded only in tandem use. With the upper clamp set at some reasonable value, and a time constant set at 10 msec, the lower value can be varied from 0 (no lower clamp) to something like 40 dB below the upper clamp, with no severe change in the output speech. What does change is the noise in speech silences. With a lower clamp of zero, all input noise is encoded producing noise output from the receiver. When the lower clamp is moved up, the noise in silences is suppressed at a reduction in dynamic range. From the point of view of tandem operation, a lower clamp close to zero allows for minimum loss of speech sounds (minimum loss of dynamic range) at a cost of some noise in silent intervals. This study indicates that CVSD and similar devices produce a speech SNR reasonably close to what this class of devices is capable of producing.

The second distinct software system was designed to investigate tandem interactions between wideband and narrowband terminals. The system allows for the storage of a raw digitized speech signal on the Univac drum. The waveform can then be played out to a speech coder, and the coder output can also be stored on the drum. In its turn, the coder output can be played out

to a second coder, with this final output also stored on the Univac drum. In this manner, a tandeming situation can be set up in a reproducible fashion with a fixed-speech utterance as a probe. The software and hardware systems as set up allow waveforms to be displayed in flexible formats, and any of the node outputs to be heard out of the D/A converter. This system was used primarily to investigate LPC output waveforms into CVSD coders. This tandem combination was indicated by T&E results to be a particularly severe one. To reduce the peaky LPC output waveform and, in turn, reduce the CVSD slope overload distortion, the LPC waveform was synthesized with a modified LPC filter. The modified filter was obtained by shifting all the filter poles closer to the unit circle in the sampled data z-plane. This was done with a simple filter scaling in the iteration, and has the effect of narrowing the bandwidth of all the poles. The technique was of limited value and introduced severe stability problems which could not be solved except with increased computational complexity. Study of several all-pass phase-dispersion filters led to modest improvement in overall tandem performance, from a waveform point of view, but seemed to make little difference in listening tests. Finally, a class of dispersive filters derived from the radar "FM chirp" concept appeared to yield some improvement in paired listening comparisons of LPC-CVSD tandem situations, with and without the dispersion filters.

Design of a (digital) chirp filter can be understood by first examining the mathematical expression for an analog chirp;

$$h(t) = \sin \frac{\pi W}{T} t^2 \qquad \text{(VII-1)}$$

where t is time, W is the chirp bandwidth, and T is the duration of the chirp. To show that W is really the bandwidth, note that $\alpha = \pi W t^2/T$ is the instantaneous phase, and therefore the instantaneous frequency

$$f = \frac{1}{2\pi} \frac{d\alpha}{dt} = \frac{Wt}{T} \quad . \qquad \text{(VII-2)}$$

Thus, when $t = T$, the frequency has chirped all the way up to W. Intuition tells us that the signal h(t) should have a spectrum that extends from about $-W$ to $+W$, but this doesn't tell us enough about the spectral details.

Digital chirp filters are not quite as well known, although there has been some discussion of their properties in a radar context.† A simple way to translate Eq. (VII-1) into a digital signal is to let $t = nT_s$ (where T_s is the sampling interval) and $T = MT_s$ where M is the number of samples in the signal; then

$$s(n) = h(nT_s) = \sin \frac{\pi W}{MT_s} n^2 T_s^2 = \sin \frac{\pi (WT_s) n^2}{M} \qquad \text{(VII-3)}$$

with the product WT_s defined as the "chirp constant." Our first task therefore is to choose (WT_s) and M for best results. For speech, W should be about 3.5 kHz, so that the chirp sweeps out the pertinent audio band. M is the parameter we have to play with; a small M will yield a very non-flat spectral response and distort the speech spectrum, while a large M tends to be expensive to implement.

† L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing (Prentice-Hall, Englewood Cliffs, New Jersey, 1975).

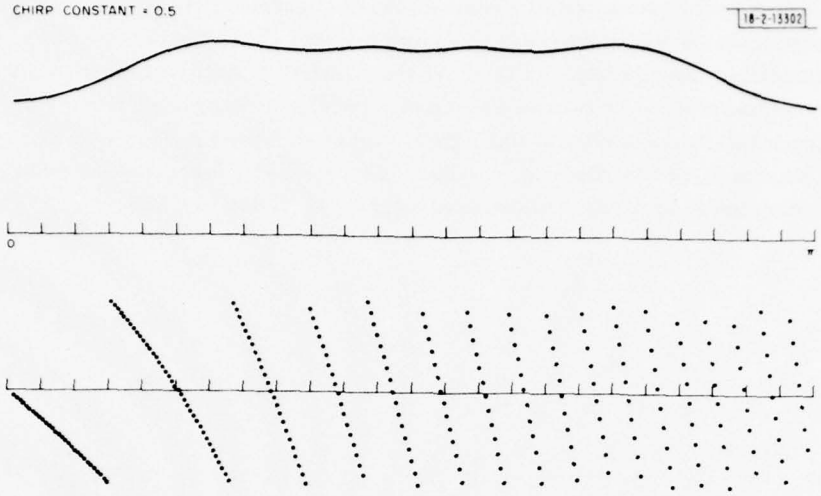


Fig. VII-1. Amplitude and phase characteristics of 34-tap chirp.

Figure VII-1 presents a compromise chirp filter in magnitude and phase. The magnitude function is presented from 0 to π, which represents the range up to half the sampling frequency. The chirp constant (WT_s) of 0.5 represents a chirp right out to half the sampling frequency. The linear-amplitude scale is down about 3 dB at both high and low ends. The phase display is that of the appropriate quadratic function for the chirp, but it is difficult to see because of the $\pm\pi$ representation. Figure VII-2 presents the impulse response of a typical chirp filter. For the given 34-tap filter, the overall delay dispersion from zero frequency up to the half-sampling frequency is about 4.25 msec.

This 34-tap filter as well as a 46- and a 64-tap version were included in the tandem demonstration of LPC-chirp filter-CVSD delivered to Reston at the end of FY 7T. The demonstration allowed for comparative listening to LPC-CVSD tandem with and without the dispersive filters.

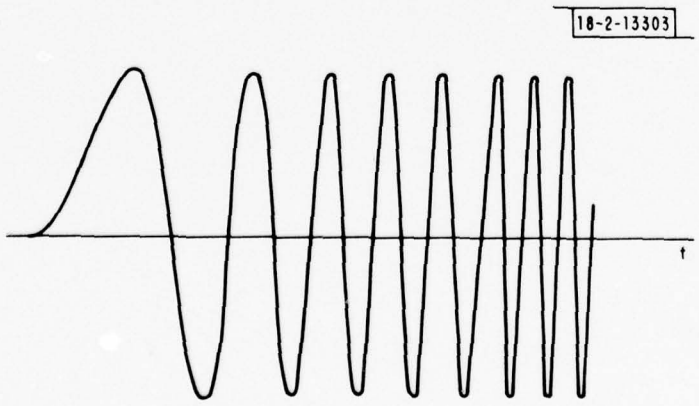


Fig. VII-2. Unit sample response of typical chirp filter.

As a result of our tandeming and wideband studies as described, it appears that an ongoing research strategy must be directed toward improved quality of both the narrow and wideband equipments separately. Any attempts to improve their interoperability seem to yield very small gains not at all comparable to the overall reduction in quality and intelligibility the tandem combinations produce. Some early work this past year on wideband APC systems produced higher-quality outputs at 16 kbps than any 16-kbps CVSD devices. However, we must reduce APC hardware complexity to produce a viable wideband alternative to CVSD.

VIII. MICROPROCESSOR REALIZATION OF A LINEAR-PREDICTIVE VOCODER

A. INTRODUCTION

For the past several years there has been a trend toward the realization of narrowband speech terminals in the form of small general-purpose digital computers. These computers have been fast enough to run the "real-time code" necessary to transform them from general-purpose computers to speech terminals capable of full duplex operation between talker-listener and modem. This approach was necessitated by the flux in narrowband speech algorithms during this time. As a result of recent work in linear predictive coding (LPC) techniques[1,2] applied to the analysis-synthesis of speech, it has become possible to specify an LPC approach[3] which produces acceptable narrowband speech in the range from 2.4 to 4.8 kbps. In addition, a recent project at Lincoln Laboratory[4] provided the opportunity to implement the pertinent LPC code, pitch-detector code, and data-handling code in a very "lean" manner in terms of program and data memory use, and efficient real-time operation. This previous experience has enabled us to approach the design of a microprocessor-based LPC vocoder with full knowledge of each subroutine and all timing sequences needed for interaction with both the incoming and outgoing audio data, as well as the outgoing and incoming digital data stream.

Our starting goal for a microprocessor-realized linear-predictive vocoder was the production of a compact, low-power, inexpensive device using commercially available integrated circuits. We were willing to design a completely special-purpose device[5] that would implement only the LPC voice terminal in an efficient form. In addition, there was no consideration of custom large-scale-integration chip use since the costs for a limited vocoder market appeared too high, and no small set of chip types seemed adequate. In effect, the goal was a benchmark device using only commercial chips whose price would drop with the larger commercial market. This benchmark device could then be used in larger system designs as a cheap building block, or could be modified and expanded to include modem and other functions.

Starting with a study of available microprocessor chip sets, a particular choice was made on the basis of speed, signal-processing power, and basic chip organization (the AMD 2900 series). Several design iterations were then made starting with a machine using three separate microprocessor CPEs. In this design each CPE was doing a special-purpose task, and was fed from separate analog-processing circuits. Because of inefficiencies associated with memory sharing and access, this design evolved to a two CPE machine which was physically divided into a transmitter and separate receiver. This design also appeared inefficient. Finally, it was seen that a single CPE and hardware multiplier could satisfy all the signal-processing requirements for the given algorithms. A complete software study then preceded the detailed logic design. In effect, all the machine code was written or blocked out to verify the design. In spite of our avowed goal of a special-purpose vocoder device, in the end we designed a rather general-purpose structure. The limited in-out capability as well as the limited data and program memory are what remain of the special-purpose device. The end design is based on a single microprocessor CPE augmented with a four-cycle multiplier. The basic structure is that of a two-bus general-purpose machine with separate program and data memory as shown in Fig. VIII-1.

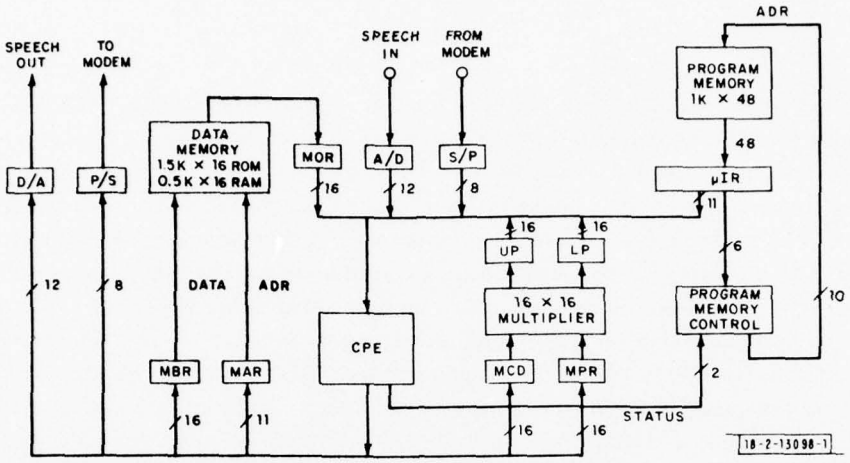


Fig. VIII-1. LPCM block diagram.

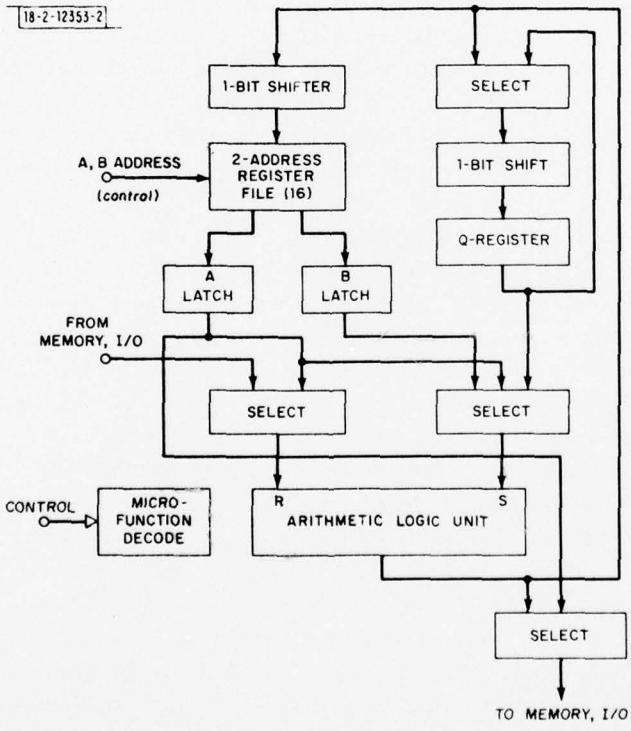


Fig. VIII-2. CPE chip block diagram.

B. LPCM SYSTEM DESCRIPTION

1. Architecture

The basic block diagram for the LPCM is shown in Fig. VIII-1. All instructions for this machine are executed in a 150-nsec cycle except the multiply which requires four machine cycles or 600 nsec. The nucleus of this system is the CPE which is based on the AMD 2901 microprocessor chip. Four such chips are used along with a carry-lookahead chip to yield a 16-bit CPE.

A simplified block diagram of the 2901 appears in Fig. VIII-2. From this diagram it can be seen that the chip consists of an ALU capable of add, subtract, and Boolean operations coupled with an internal 2-port general register file consisting of 16 words. Multiplexers at the input of this register file permit a 1-bit up-or-down shift prior to writing the memory. A Q-register is provided which allows double precision shifts to be implemented. Inputs to the chip from the outside world consist of two 4-bit addresses for the internal register file, control signals, and data from external devices such as memory or I/O devices. The manufacturer's literature should be consulted for further details about the 2901.

Referring again to Fig. VIII-1, it is seen that the 16-bit CPE is connected to an input and an output data line. The input line is multiplexed between 6 data sources, the 16-bit memory output register (MOR) of the data memory, the 12-bit A/D converter, the 8-bit serial-to-parallel (S/P) converter, the 16-bit upper and lower products coming from the multiplier, and an 11-bit field coming from the instruction register. The data memory consists of 2K 16-bit words, 1.5K of which are ROM and contain the various lookup tables needed to implement the LPC algorithm. The output of the CPE is channeled to the D/A converter, the parallel-to-serial (P/S) converter, the memory buffer and address registers (MBR and MAR), and the multiplicand (MCD) and multiplier (MPR) registers of the multiplier. These various output registers are clocked under the control of a 3-bit field in the instruction register.

The multiplier uses the Booth-McSorley algorithm to multiply two 16-bit two's-complement numbers and makes the full 32-bit product available to the CPE's input ports in two 16-bit pieces. The multiplier is fabricated from the AMD25S05 4×2 multiplier chip. Eight of these are used to construct a 16×4 array multiplier which is clocked four times to yield the final product. The outputs are fully buffered so that the product may be retrieved from the multiplier any time four machine cycles or longer after the start of the multiply. The CPE is free to do other tasks in this interval while multiplication is taking place.

The program memory contains 1K of 48-bit words. The output of this memory is clocked into a micro-instruction register, and the memory address is derived from the program control logic. The latter is based on the AMD2909 program sequencer chip, a simplified block diagram of which appears in Fig. VIII-3. Three of these 4-bit chips are used, making it possible to address 4K of program memory even though only 1K of such memory is needed for the present application. The 2909 controller is driven by a 2-bit control line which enables one to select the next program address to be either the last address plus one, a jump address which comes from the micro-instruction register, the latest address on the internal stack, or an interrupt address determined by the I/O system. The jump logic which drives the control ports of the 2909 allows for unconditional jumps, conditional jumps depending on the status bits coming from the CPE, and jumps to and returns from subroutines. Subroutines may be nested up to four deep when interrupts are locked out, and three deep when they are active.

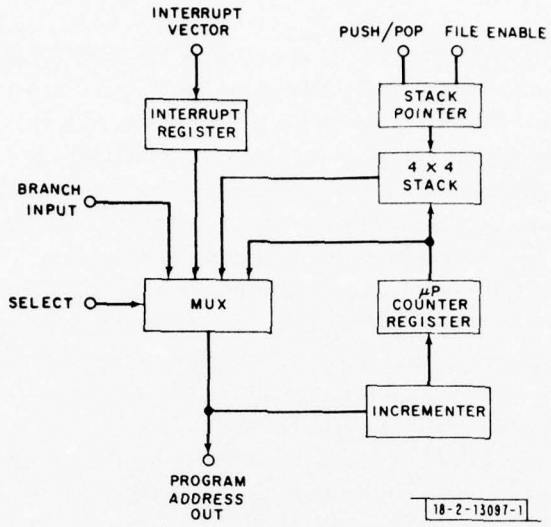


Fig. VIII-3. Program sequencer chip block diagram.

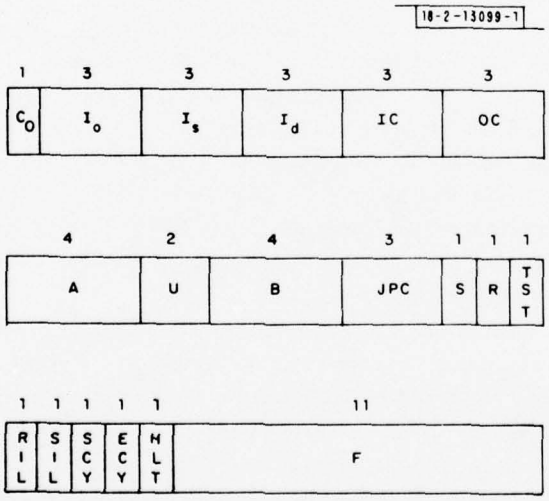


Fig. VIII-4. LPCM micro-instruction word format.

The I/O system for the LPCM consists of two input channels – the A/D and S/P converters – and two output channels – the D/A and P/S converters. The A/D – D/A channels run on a common 129.6-μsec clock that is derived from the 150-nsec system clock. The P/S and S/P converters run on external modem clocks which must have the same nominal frequency (2400, 3600, or 4800 Hz), but which may be asynchronous to one another. The I/O channels generate an interrupt request whenever their associated clocks present a rising edge to the system. This request causes the program control logic to produce a jump to one of three predetermined locations in program memory at the first instance the system finds itself in a position to allow interrupts. Several interrupts may have requests pending at one time; they are serviced in order of their priorities which are P/S, S/P, and A/D – D/A. While a given interrupt is being serviced, all others are locked out. Upon return from an interrupt service routine, the software releases interrupt lockout thus enabling the honoring of further interrupt requests.

2. Instruction Format

The format of the 48-bit-wide instruction word is shown in Fig. VIII-4. The instruction word is divided into various fields of varying length, the functions of which will now be discussed.

The C_o, I_s, and I_o fields determine the basic operation that the CPE is to perform, e.g., add the contents of internal register at address A to the contents of the internal register at address B, or take the external data presented to the chip and logically AND them with the contents of the internal register at address A. A list of useful combinations of these fields, along with a mnemonic for each, is given in Appendix A at the end of this Section.

The I_d field determines where on the CPE chip the output of the ALU is to go. Some examples are: the output of the CPE alone, the output of the CPE and internal register file at address B, or the output of the CPE and the Q-register.

The IC and OC fields determine where the CPE gets its input and where its output is to go, respectively. The IC field steers the input 6-way multiplexer to any of the input sources mentioned above, and the OC field determines which, if any, of the output registers connected to the CPE are to be clocked. The A and B fields simply supply the addresses to the CPE's two-port memory and need no further discussion.

The JPC field along with the R and S fields provides program control by means of various kinds of jumps. A complete list of these appears in Appendix A. Conditional jumps in the LPCM are somewhat unconventional in that the condition on which the jump is to be based must be established in an instruction preceding the actual jump instruction by means of the TST field. More precisely, if one wishes to conditionally jump, say, based on whether one of the CPE's internal registers is zero, then the contents of this register must be made to appear at the CPE output with an instruction that also has the TST bit set. This strobes the CPE status into a (2-bit) status register which, in turn, may be tested by a subsequent instruction containing the appropriate jump code.

The remaining fields are quite straightforward. The F field appears directly at the CPE input where it can be used for a constant or a base address. This field also contains the jump address and must be set accordingly for each instruction containing a jump. The SIL and RIL fields are used to set interrupt lockout and release interrupt lockout, respectively, and are primarily used to prevent interrupts while executing calculations that an interrupt could destroy such as an ongoing multiply. The SCY and ECY fields are provided to facilitate multiple-precision adds and subtracts. When the SCY bit is set during an add or subtract instruction,

the carry resulting from this operation is saved in a flip-flop. This saved carry can then be used in a later add or subtract instruction by setting the ECY bit during that instruction. Finally, the HLT bit stops the machine – a feature that is only used during debugging operations. The two bits labeled U are unused.

3. Data-Memory Addressing

Addresses for the LPCM data memory must be generated in the CPE and then deposited in the MAR. Direct addressing of data memory is achieved by having the desired address in the F field of the micro-instruction word and passing it through the CPE to the MAR. Indexed addressing can be accomplished by having a base address in the F field, adding to it the contents of a CPE internal register, and depositing the result in the MAR. It should be noted, however, that the contents of the addressed location in data memory are only available as a CPE input one instruction cycle after the desired address is placed in the MAR. This is due to the fact that the memory output is buffered in the MOR. Writing data memory is also a 2-step process in the sense that the address must first be calculated and deposited in MAR before the datum itself may be read out into the MBR.

4. *Timing Considerations*

The basic events that must take place in order to execute an LPCM instruction are:

(a) Program counter assumes desired state
(b) Program memory is accessed
(c) Accessed instruction is executed by CPE.

It is not possible to perform all three of these operations in the desired cycle time of 150 nsec, so the sequence is broken into two parts by inserting the microprogram instruction register after the program memory. This results in what is called a doubly overlapped pipeline structure in which instruction fetch takes place in parallel with execution of the instruction fetched on the previous machine cycle. This type of pipelining is transparent to the programmer of the LPCM.

The LPCM also employs pipelining in the data memory acquisition path and in the jump control path as has been described earlier. This pipelining is not transparent to the programmer in that memory addresses and jump conditions must be set up sufficiently in advance of the instruction that makes use of them. Experience has shown that careful programming can usually circumvent any potential loss of program efficiency caused by these pipelined paths in the machine.

C. ENGINEERING CONSIDERATIONS

The present LPCM is a prototype designed to demonstrate that a dedicated linear predictive vocoder can be realized both cheaply and compactly using off-the-shelf components. Since it is a prototype, it was decided to use standard 16- × 7-in. universal wirewrap boards as the packaging medium rather than go directly to smaller PC boards. Universal boards were chosen because the LPCM uses every standard package size from 14- to 40-pin in its design. The final design uses 162 DIPS and occupies 1.5 boards. These figures include all the analog circuits required before and after the A/D and D/A converters. The power consumption of the device is less than 45 W. A photograph of the completed LPCM appears in Fig. VIII-5.

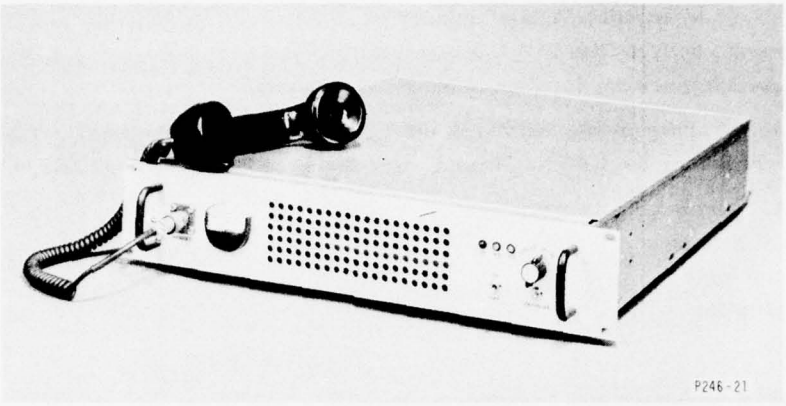


Fig. VIII-5. The completed LPCM.

Appendix B at the end of this Section gives a complete compilation of the parts used to fabricate the LPCM. Included in the table are military and commercial cost figures for building 1, 500, 1000, and 10,000 processors. These figures are based on the extrapolation rules provided by the Narrow Band Voice Consortium Subcommittee for estimation of "cost to produce." The figures referring to the packaging of the LPCM are estimates of how it could be packaged using PC boards, and do not reflect the present wirewrap packaging of the prototype.

D. DEBUGGING AND TEST SYSTEM

1. *Hardware and Software Debugging Aids*

The LPCM is intended to be a stand-alone device with its control program residing in PROMs. During the debugging phase, however, it is necessary to replace the PROM with RAM in order to facilitate program changes and allow the running of diagnostic programs. In addition, it is extremely advantageous to have a means for starting and stopping the machine, setting breakpoints, and examining the contents of data memory and the CPE's internal register file.

The above requirements were met by the design and fabrication of a separate unit – the LPCM tester – which is connected to the LPCM by means of cables during the debugging phase. The main component of the tester is a 1024×48 RAM which effectively replaces the PROM destined to reside in the LPCM. In addition, the tester duplicates the AM2909 program control chips that are located in the LPCM itself. This was done to minimize both the number of control cables between the LPCM and its tester and the tester-oriented logic needed in the LPCM.

The tester's program memory can be loaded in either of two ways: (a) one register at a time by means of front-panel switches, or (b) the entire memory can be loaded from a host computer. The first mode is useful for toggling in small test programs and for patching larger programs. The latter mode is used for loading large programs such as the diagnostic system or the LPC vocoder program itself. When the tester is connected to the LPCM, the following control functions are available:

(a) Start program at an arbitrary address
(b) Stop program
(c) Single-step program

(d) Stop at breakpoint determined by switches
(e) Inspect any location in data memory
(f) Inspect any location in CPE register file
(g) Inspect/change any location in program memory.

In addition to the above-mentioned hardware debugging aids, an extensive software diagnostic system was written for the LPCM. This system tests the following functions of the LPCM:

(a) RAM portion of data memory
(b) CPE functions
(c) Jump logic
(d) Multiplier
(e) I/O.

2. The LPCM Simulator and Assembler

A simulator for the LPCM was written on a Univac 1219 computer so that software debugging could take place in parallel with the fabrication of the LPCM hardware. The simulator accepts as its input the binary code generated by an LPCM assembler. This assembler was also written on the Univac 1219 and is a straightforward two-pass assembler that understands LPCM mnemonics and symbolic addresses. Symbolic code is generated using the Univac's editor and then fed to the assembler which produces a binary output that can be loaded into the LPCM or operated on by the simulator. This same binary output was later used to burn in the PROMs that comprise the LPCM's program memory.

The simulator is fairly sophisticated in that it simulates all I/O operations, including interrupts. This allowed the debugging of not only the diagnostic package but the entire LPC vocoder program itself. In the final stages of the vocoder programming, real speech was used as the input to the simulator and the synthetic-speech output of the program was stored on magnetic tape. All computation was done in non-real time, but the final output tape was then played back in real time to provide convincing evidence that the LPCM vocoder algorithm was functioning correctly. This indeed proved to be the case, because only a few additional program bugs were found when the program was finally running on the LPCM itself.

E. FIRMWARE CONSIDERATIONS

1. The LPC Algorithm

LPC was first described by Atal and Hanauer in 1971.[1] Since then, many variations on this algorithm have appeared in the literature (see bibliographies in Refs. 2 and 6). We have chosen to implement the Markel form of the LPC algorithm for reasons detailed in Ref. 7.

This algorithm is described in block-diagram form in Fig. VIII-6. Speech samples taken every 129.6 μsec are divided into 158-point nonoverlapping groups corresponding to approximately 20 msec of data. These groups are multiplied by a Hamming window and then used to form $P + 1$ autocorrelation coefficients $R_0, \ldots, R_P$. The parameter P is the order of the filter used to model the vocal tract, and ranges from 10 at 2400 bps to 12 at 3600 and 4800 bps.

The autocorrelation coefficients are used as the constants in a set of linear equations that must be solved to obtain the parameters of the vocal-tract filter. These equations are solved by means of the Levinson recursion[8] which yields a set of P reflection coefficients $K_0, \ldots, K_{P-1}$ and a residual energy E. These reflection coefficients will be used at the receiver to implement

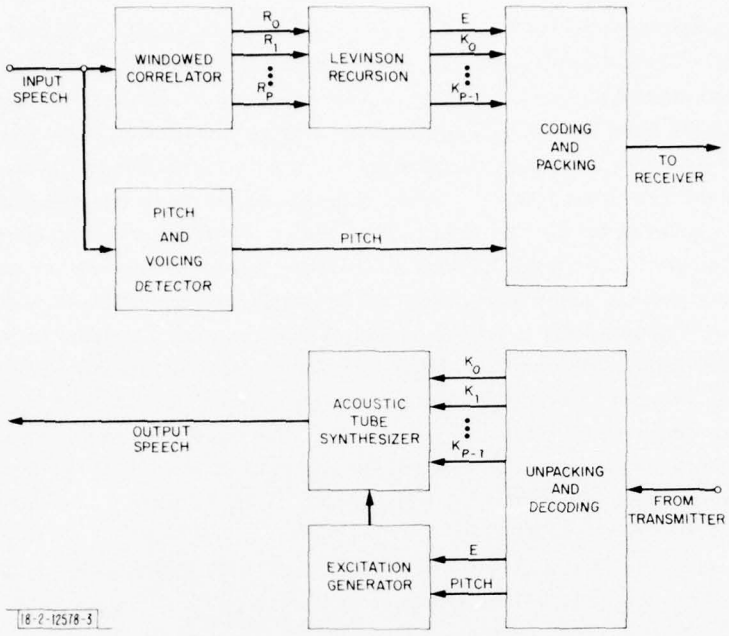


Fig. VIII-6. The LPC vocoder algorithm.

the vocal-tract filter. The structure chosen for this filter is the acoustic-tube filter described in detail in Ref. 2. The residual energy is used at the receiver to generate the amplitude of the excitation for the acoustic tube.

In addition to the processing described above, the raw speech samples are fed to a pitch and voicing detector which produces both a voiced-unvoiced decision and an estimate of pitch. The particular algorithm used for this purpose is the Gold-Rabiner pitch detector which is described in detail in Refs. 9 and 10.

The parameters produced as described above are next coded and formed into a serial bit stream for transmission to the remote receiver. The receiver portion of the algorithm accepts such a serial bit stream from the remote transmitter and unpacks it to form the code-book addresses of the various parameters. These addresses are then decoded to obtain the actual values of the parameters, which are then used to implement the acoustic-tube filter and its excitation. The output of the filter is the final synthetic speech.

The coding of the parameters, except for pitch which is transmitted as is, is accomplished by a logarithmic-search table-lookup routine. The residual energy is logarithmically coded to 5 bits. The reflection coefficients are coded by means of truncated, log-area ratios. Each reflection coefficient is first clamped to an individually selected interval, transformed by the log-area-ratio function $\{\log[(1 - K)/(1 + K)]\}$, and finally truncated to the desired number of bits. The number of bits used for the individual K's is a function of the desired transmission rate.

2. Implementation of the LPC Algorithm

The LPC program consists of four major pieces – a background program that handles all the computation that need only be performed once per frame and three interrupt-service routines that handle the computations that must be done for each modem clock and each A-D/D-A clock.

The A-D/D-A interrupt-service routine uses the newly arrived speech sample to update the current windowed correlation and the six elementary pitch detectors. In addition, the acoustic-tube filter is updated to produce a new synthetic-speech sample for the D/A converter. This approach eliminates the need for any substantial buffering of raw speech, thus reducing our data memory requirements. The reflection coefficients for the acoustic tube are interpolated against the coefficients for the next frame every 5 msec, and the amplitude is interpolated every time a new pitch pulse is generated. No amplitude interpolation takes place during unvoiced frames.

The main task of the P/S converter interrupt-service routine is to pass the coded data produced by the analyzer portion of the program to the transmit modem. This is accomplished by loading the first code word into the P/S converter and then counting a number of interrupts equal to the known number of bits in this word. Subsequent words are then loaded and the appropriate number of interrupts counted after each. When a complete frame of code words has been serialized in this fashion and passed to the transmit modem, the current correlation coefficients are transferred to registers used by the background routine, the correlator is reset to start a new correlation, and a flag is set to tell the background routine to start a new frame calculation using the new correlation coefficients.

The S/P interrupt-service routine receives serial data from the receiver modem. It deserializes this stream into the proper-length code words using an interrupt-counting technique similar to the one used by the P/S converter. The code words are then used to access decoding tables thus producing the parameters eventually used by the acoustic-tube synthesizer. These parameters are transferred to the buffer used by the acoustic tube when the S/P routine's counters determine that it has received a complete frame of new data.

The deserialization procedure just described only makes sense if the S/P routine "knows" where the first code word of a frame is in the incoming bit stream. The process of making this determination is known as frame synchronization and is another task of the S/P routine. Frame synchronization is established by having the transmitter transmit a known bit pattern in place of the pitch word during unvoiced utterances. The pattern is chosen to correspond to an illegal (too high) pitch so that the receiver can still make an unambiguous buzz/hiss decision. The frame synchronization algorithm now consists simply of searching for this known pattern in the serial bit stream as it arrives at the receiver. Synchronization is declared (i.e., knowledge of the location of the pitch word) when, and only when, the known pattern has been found at the same location in six consecutive frames. When this occurs, the S/P routine sets its bit and word counters accordingly, thus establishing synchronization.

The final routine to be discussed is the background routine. The start of this routine is an idle loop whose sole purpose is to continually check the status of the frame-ready flag that is set by the P/S interrupt-service routine. As long as this flag is clear, the program remains in the idle loop except for those times when an interrupt arrives and transfers control to the appropriate service routine. When the flag is finally set, the program drops out of the idle loop and begins its once-a-frame computations. The first of these is the final determination of pitch by a routine that examines the status of the six elementary pitch detectors and produces a buzz/hiss decision and an appropriate pitch. Next, the double-precision correlation coefficients are put into a block-floating-point format based on R(0) and passed on to the Levinson recursion which produces the desired reflection coefficients and the residual energy. The latter is unnormalized to remove the scale factor introduced by the block-floating-point routine, and then the parameters are coded using the appropriate coding tables. The final code words are placed in a buffer where

the P/S routine can access them for shipment to the transmit modem. Control is then returned to the idle loop. It should be emphasized that, while the background routine is calculating, interrupts are active which means that the background routine is only actually working in the intervals when no interrupt-service routine is in progress.

One final routine should be mentioned, namely the initialization routine. This routine starts at program address zero and is only entered on power-up or when the initialize pushbutton is pressed. The main function of this routine is to clear data RAM, initialize the few RAM registers that require it, and finally determine which rate vocoder is desired. The latter function is accomplished by sensing a front-panel rate-control switch and then setting pointers to the proper coding and decoding tables. In addition, if the rate selected is 2400 bps, the filter order is changed from 12 to 10.

F. CONCLUSIONS

The design concessions that mark the LPCM as a special-purpose machine designed to be a speech terminal are: limited I/O capability, and limited data and program memory. The I/O bus only communicates with A/D – D/A, parallel-to-serial modem input, and serial-to-parallel modem output. The LPCM data memory consists of 1536 locations of 16-bit ROM tables and 512 locations of 16-bit RAM words. The program memory consists of 1K by 48 bits of ROM, of which less than 800 locations are used. A priori knowledge of the operating algorithms as well as an operating simulator and diagnostics reduced the entire time from design to completion to less than one year. The present package requires 162 DIPs including audio circuits, dissipates less than 45 W, and occupies about 1/3 cubic foot. The operating code occupies the machine for about 65 percent of real time.

As a prototype device, the LPCM specifications are not as tight as they might be. Given the 65-percent utilization, the cycle time can be slowed to over 200 nsec and power dissipation reduced by roughly 10 W. The volume can be reduced by as much as a factor of 3 if PC boards are used and tighter packaging is designed.

The overall package count of 162 various-sized DIPs includes the 7 packages of AMD [CPE (4) and AMD sequencer (3)], about 40 packages of memory and memory-related circuits (20 packages for multiplier, and the rest for I/O), bus multiplexing, timing, interrupt, and branching. It is clear that in terms of power and size the device is not defined by the microprocessor chips. The overall machine size is determined by all of the "glue logic" and memory packages which swamp out the microprocessor chips. In fact, the memory and memory-related packages probably represent a lower bound on size and power, in the sense that everything else may shrink considerably, but the current memory size and power are relatively static.

APPENDIX A: LPCM MNEMONICS

The following is a compilation of the bit assignments that must be made to the fields of the LPCM micro-instruction word to achieve various functions. Each of these assignments is preceded with a mnemonic that can be used when preparing code for the LPCM assembler. The first group of these assignments are the so-called "op codes" which affect the C_o, I_o, and I_s fields. The format of the presentation consists of a mnemonic followed by a 3-digit octal number giving the values assigned to C_o, I_o, and I_s, respectively, followed by a brief description of

the operation accomplished by the assignment. The result of the operation appears at the internal ALU output port. The following notations are used in the descriptions:

R(A)	Contents of internal register addressed by the A field.
R(B)	Contents of internal register addressed by the B field.
Q	Contents of the Q-register.
D	Data at input port of the CPE
·	Logical AND
!	Logical OR
⊕	Logical exclusive OR
%	Logical complement

It should be noted that all possible operations which the CPE is capable of are not included in the following list.

ADDAB	001	R(A) + R(B)
ADDDA	005	D + R(A)
ADDAB1	101	R(A) + R(B) + 1
ADDDA1	105	D + R(A) + 1
SUBBA	111	R(B) − R(A)
SUBAD	115	R(A) − D
SUBAB	121	R(A) − R(B)
SUBDA	125	D − R(A)
SUBBA1	011	R(B) − R(A) − 1
SUBAD1	015	R(A) − D − 1
SUBAB1	021	R(A) − R(B) − 1
SUBDA1	025	D − R(A) − 1
MOVB	033	R(B)
MOVA	034	R(A)
MOVD	037	D
INCB	103	R(B) + 1
INCA	104	R(A) + 1
INCD	107	D + 1
DECB	013	R(B) − 1
DECA	014	R(A) − 1
DECD	027	D − 1
CSB	123	−R(B)
CSA	124	−R(A)
CSD	117	−D
ANDAB	041	R(A) · R(B)
ANDDA	045	D · R(A)
ORAB	031	R(A) ! R(B)
ORDA	035	D ! R(A)
XORAB	060	R(A) ⊕ R(B)
YORDA	065	D ⊕ R(A)

CMPB	023	%R(B)
CMPA	024	%R(A)
CMPD	017	%D
CLR	142	0

The next set of assignments concerns the destination field I_d, which determines where the output of the ALU is to go. The format is mnemonic, 1-digit octal number, and description. The notations F for ALU output and Y for CPE output are used in the descriptions.

Q	0	F → Q, F → Y
Y	1	F → Y
RAY	2	F → R(B), R(A) → Y
R	3	F → R(B), F → Y
SDD	4	Double-precision down shift [F, Q]/2 → [R(B), Q] F → Y
SD	5	F/2 → R(B), F → Y
SUD	6	Double-precision up shift [F, Q]* 2 → [R(B), Q] F → Y
SU	7	F* 2 → R(B), F → Y

The next set of assignments concerns the IC field which controls the input multiplexer to the CPE. The format is mnemonic, 1-digit octal number, and description.

SP	0	Serial-to-parallel converter
ADC	1	A/D converter
LP	2	Bits 0 to 15 of the product
UP	3	Bits 15 to 30 of the product
MOR	4	Memory output register
FD	5	11-bit instruction field

The clocking of the various registers connected to the output of the CPE is controlled by the output control field OC. The format is the same as for the input control field.

NIL	0	Clock nothing
MAR	1	Clock memory address register
MBR	2	Clock memory buffer register
MCD	3	Clock multiplicand register
DAC	4	Clock D/A converter buffer register
PS	5	Clock into P/S converter
MPR	6	Clock multiplier register and start multiply sequence

The final group of assignments concerns the jump control fields: JPC, S, and R. The format is mnemonic, 3-digit octal numbers giving the assignment to the JPC, S, and R fields, respectively, and a description.

NIL	000	No jump
JP	100	Unconditional jump
JPZ	200	Jump if positive or zero
JZ	300	Jump if zero

JN	400	Jump if negative
JNZ	500	Jump if not zero
JSW	600	Jump if switch w on
JSV	700	Jump if switch v on
JPS	110	Unconditional jump to subroutine
JPZS	210	Jump to subroutine if positive or zero
JZS	310	Jump to subroutine if zero
JNZS	410	Jump to subroutine if negative
JSWS	610	Jump to subroutine if switch w set
JSVS	710	Jump to subroutine if switch v set
SBR	101	Return from subroutine

APPENDIX B: LPCM SPECIFICATIONS

Cycle Time	150 nsec	
Basic Logic Family	TTL using low-power Schottky TTL in AMD chips, high-power Schottky where necessary in critical paths.	
Program Memory (ROM)	1K × 48 bits	12 – MMI 6351 (1K × 4)
Data Memory (ROM)	1536 × 16 bits	4 – MMI 6351 (1K × 4) 2 – FCLD 93448 (512 × 8)
Data Memory (Active)	512 × 16 bits	8 – FCLD 93442 (256 × 4)
Hardware Multiplier	One quarter of an array operating in 150-nsec 4 × 16 multiply	8 – AMD 25S05 (2 × 4)
Basic CPE		4 – *AMD 2901 (4-bit slice)*
Microsequencer		3 – AMD 2909 (4-bit slice)
Audio Conditioning	12-bit A/D, D/A conversion at 129.6-μsec samples. Input Filter 8th order, elliptic filter 52-dB stop-band attenuation 1.2-dB ripple, cutoff at 3596 Hz. Output Filter 8th order, elliptic filter 41-dB stop-band attenuation 0.2-dB ripple, cutoff at 3596 Hz.	
Total DIP Count	162	
Total Power Dissipation	45 W	
Construction Technique	Two universal wirewrap boards (50 percent of second board unused) 7 × 16 in. Center plane voltage Two outside planes ground	

AD-A041 246 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 17/2
SPEECH EVALUATION.(U)
SEP 76 B GOLD
UNCLASSIFIED F19628-76-C-0002
ESD-TR-76-382 NL

2 OF 2
ADA041246

END
DATE FILMED
8-77

ITEM	QUANT. PER UNIT	SOURCE	ITEM COST		1 PROCESSOR			500 PROCESSORS			1,000 PROCESSORS			10,000 PROCESSORS		
			Comm.	Mil.	Quant. Mult.	Comm.	Mil.	Quant. Mult.	Comm.	Mil.	Quant. Mult.	Comm.	Mil.	Quant. Mult.	Comm.	Mil.
7400	6	TI	0.86	3.27	6	5.16	19.62	3.96	3.18	12.12	2.82	2.43	9.22	2.10	1.80	6.86
7442	1	TI	3.05	9.16	1	3.05	9.16	0.66	2.01	6.05	0.66	3.71	6.05	0.35	1.07	3.21
7408	2	TI	0.98	3.53	2	1.96	7.06	1.32	1.29	4.66	0.94	0.92	3.32	0.70	0.69	2.47
7432	1	TI	1.73	4.65	1	1.73	4.65	0.66	1.14	5.30	0.66	1.14	5.30	0.35	0.61	1.63
7404	9	TI	1.41	4.04	9	12.69	36.36	4.23	5.96	17.09	4.23	5.96	17.09	3.15	4.44	12.73
7410	2	TI	1.14	3.27	2	2.28	6.54	1.32	1.50	4.32	0.94	1.07	3.07	0.70	0.80	2.29
74S151	1	TI	3.04	22.11	1	3.04	22.11	0.66	2.01	14.59	0.66	2.01	14.59	0.35	1.06	7.74
74S257	4	TI	3.97	21.24	4	15.88	84.96	1.88	7.46	39.93	1.88	7.46	39.93	1.40	5.56	29.74
74S00	2	TI	1.39	8.15	2	2.78	16.30	1.32	1.83	10.76	0.94	1.31	7.66	0.70	0.97	5.70
74S02	6	TI	0.88	8.15	6	5.28	48.90	3.96	3.24	30.24	2.82	2.48	22.98	2.10	1.84	17.12
74S04	5	TI	1.69	10.12	5	8.45	50.60	3.30	5.58	33.40	2.35	3.97	23.78	1.75	2.96	17.71
74S74	2	TI	2.79	14.55	2	5.58	29.10	1.32	3.68	19.21	0.94	2.62	13.67	0.70	1.95	10.19
74S174	25	TI	5.55	33.90	25	138.75	847.50	8.75	48.56	296.63	8.75	48.56	296.63	8.75	48.56	296.63
74S157	1	TI	3.75	21.24	1	3.75	21.24	0.66	2.48	14.87	0.66	2.48	14.87	0.35	1.31	7.43
7414	1	TI	6.72	12.92	1	6.72	12.92	0.66	4.16	7.99	0.66	4.16	7.99	0.35	2.35	4.52
74LS74	2	TI	1.58	2.06	2	2.16	4.12	1.32	2.09	2.72	0.94	1.49	2.72	0.70	1.11	1.44
74S112	1	TI	2.02	16.00	1	2.02	16.00	0.66	1.12	9.90	0.66	1.12	9.90	0.35	0.71	5.60
74125	2	TI	1.80	3.49	2	3.60	6.98	1.32	1.11	4.32	0.94	1.69	4.32	0.70	1.26	2.38
74S253	8	Fchld.	3.75	3.75	8	30.00	30.00	5.28	19.80	19.80	3.76	14.10	14.10	2.80	10.50	10.50
74LS258	2	TI	3.94	5.12	2	3.94	10.24	1.32	2.60	7.04	0.94	3.70	7.04	0.70	2.75	3.58
74S260	2	Sig.	1.32	1.32	2	2.64	2.64	1.32	1.74	1.74	0.94	1.24	1.24	0.70	0.92	0.92
74367	2	TI	1.80	1.80	2	3.60	3.60	1.32	1.11	1.11	0.94	1.69	1.69	0.70	1.26	1.26
74S195	5	TI	6.00	25.11	5	30.00	125.55	3.30	19.80	82.86	2.35	14.10	59.01	1.75	10.50	43.94
25S05	8	AMD	19.50	33.15	8	156.00	265.20	5.28	102.96	175.03	3.76	73.32	124.64	2.80	54.60	92.82
2902	1	TI	5.67	11.40	1	5.67	11.40	0.66	3.74	7.52	0.66	3.74	7.52	0.35	1.98	3.99
74LS175	4	TI	5.25	27.29	4	21.00	109.16	1.88	9.87	51.31	1.88	9.87	51.31	1.40	7.35	38.21
74LS174	8	TI	4.19	22.36	8	33.52	178.88	3.76	15.75	84.07	3.76	15.75	84.07	2.80	32.84	64.00

ITEM	QUANT. PER UNIT	SOURCE	ITEM COST		1 PROCESSOR			500 PROCESSORS			1,000 PROCESSORS			10,000 PROCESSORS		
			Comm.	Mil.	Quant. Mult.	Comm.	Mil.	Quant. Mult.	Comm.	Mil.	Quant. Mult.	Comm.	Mil.	Quant. Mult.	Comm.	Mil.
74393	2	TI	5.49	5.49	2	10.98	10.98	1.32	7.25	7.25	1.32	7.25	7.25	0.70	3.84	3.84
74164	1	TI	4.00	21.82	1	4.00	21.82	0.66	2.64	14.40	0.66	2.64	14.40	0.35	1.40	7.64
74166	1	TI	7.00	21.82	1	7.00	21.82	0.66	4.62	14.40	0.66	4.62	14.00	0.35	2.45	7.64
8T15	1	Sig.	5.20	5.20	1	5.20	5.20	0.66	3.43	3.43	0.66	3.43	3.43	0.35	1.82	1.82
8T16	2	Sig.	7.45	7.45	2	14.90	14.90	1.32	9.83	9.83	0.94	7.00	7.00	0.70	5.22	5.22
IC PARTS TOTAL						553.33	2055.51		303.54	1013.92		257.01	900.19		216.48	720.77
MEMORY																
93422	8	Fcld.	44.90	78.62	8	359.20	628.96	5.28	237.07	415.11	3.76	168.82	295.61	2.80	125.72	220.14
93448	2	Fcld.	33.44	56.85	2	66.88	113.70	1.32	44.14	75.04	1.32	44.14	75.04	0.94	31.43	53.44
6351	16	MMI	30.00	55.00	16	480.00	880.00	7.52	225.60	413.60	7.52	225.60	413.60	5.60	168.00	308.00
MICROPROCESSOR CHIPS																
2901	4	AMD	60.00	240.00	4	240.00	960.00	2.64	158.40	633.60	2.64	158.40	633.60	1.88	112.80	451.20
2909	3	AMD	42.12	169.28	3	126.36	507.84	1.98	83.40	335.17	1.98	83.40	335.17	1.41	59.39	238.68
DIGITAL ICs TOTAL						1825.77	5146.01		1052.14	2886.44		937.37	2653.21		713.82	1992.23
MISCELLANEOUS																
A/D	1		100.00	200.00	1	100.00	200.00	0.57	57.00	114.00	0.57	57.00	114.00	0.32	32.00	64.00
D/A	1		80.00	160.00	1	80.00	160.00	0.57	45.60	91.20	0.57	45.60	91.20	0.32	25.60	51.20
S/H	1		50.00	100.00	1	50.00	100.00	0.57	28.50	57.00	0.57	28.50	57.00	0.32	16.00	32.00
Handset	1		40.00	80.00	1	40.00	80.00	0.57	22.80	45.60	0.57	22.80	45.60	0.32	12.80	25.60
Low-Pass	2		50.00	100.00	2	100.00	200.00	1.14	57.00	114.00	1.14	57.00	114.00	0.64	32.00	64.00
Osc.	1		35.00	70.00	1	35.00	70.00	0.57	19.95	39.90	0.57	19.95	39.90	0.32	11.20	22.40
Capac.	25		0.40	0.80	25	10.00	20.00	0.08	3.20	6.40	0.08	3.20	6.40	0.08	3.20	6.40
Fan	2		25.00	50.00	2	50.00	100.00	1.32	33.00	66.00	0.94	23.50	47.00	0.70	17.50	35.00
Power	1		70.00	140.00	1	70.00	140.00	0.66	46.20	92.40	0.66	46.20	92.40	0.47	32.90	65.80
PC Board	160 $in.^2$		160.00	160.00	1	160.00	160.00	0.66	105.60	105.60	0.66	105.60	105.60	0.47	75.20	75.20
Package	0.2 ft^3		400.00	600.00	1	400.00	600.00	0.57	228.00	342.00	0.57	228.00	342.00	0.32	128.00	192.00
Connector	4		6.37	18.00	4	25.48	72.00	2.28	14.52	41.04	2.28	14.52	41.04	1.28	8.15	23.04
PARTS COST						2946.25	6998.01		1694.76	4001.58		1589.24	3749.35		1108.37	2648.47
MANUFACTURING COST						8838.75	20994.03		4067.42	9603.79		3496.33	8248.57		2061.57	4105.75

REFERENCES

1. B. S. Atal and S. L. Hanauer, J. Acoust. Soc. Am. 50, 637 (1971).
2. J. D. Markel and A. H. Gray, Jr., Linear Prediction of Speech (Springer-Verlag, New York, 1976).
3. E. M. Hofstetter et al., "Vocoder Implementations on the Lincoln Digital Voice Terminal," EASCON '75, Washington, D. C., 29 September – 1 October 1975.
4. P. E. Blankenship, "LDVT: High Performance Mini-Computer for Real-Time Speech Processing," EASCON '75, Washington, D. C., 29 September – 1 October 1975.
5. P. E. Blankenship, "Preliminary Investigation of Digital Speech Processor Hardware Implementations," Technical Note 1975-8, Lincoln Laboratory, M.I.T. (5 February 1975), DDC AD-A007062/3.
6. J. D. Markel and J. J. Wolf, "Linear Prediction and the Spectral Analysis of Speech," BBN Report No. 2304, Bolt, Beranek and Newman Inc., Cambridge, Massachusetts (August 1972).
7. J. D. Markel and A. H. Gray, Jr., IEEE Trans. Acoust., Speech, and Signal Processing ASSP-22, 124 (1974).
8. N. Wiener, Extrapolation, Interpolation and Smoothing of Stationary Time Series (The Technology Press and J. Wiley and Sons, New York, 1957), Appendix B.
9. B. Gold and L. R. Rabiner, J. Acoust. Soc. Am. 46, 442 (1969).
10. M. L. Malpass, "The Gold-Rabiner Pitch Detector in a Real-Time Environment," EASCON '75, Washington, D. C., 29 September – 1 October 1975.

IX. CHARGE-TRANSFER-DEVICE IMPLEMENTATION OF CHANNEL VOCODERS

As a result of T&E comparisons between the U.K. Belgard 2.4-kbps channel vocoder and 2.4-kbps LPC systems which indicated the Belgard equipment to be competitive in areas of "robustness" and quality acceptance, a small simulation study was performed at Lincoln Laboratory. The result of the study was a Belgard simulation running on a Lincoln Laboratory DVT. Unfortunately, the channel-vocoder structure requires many digital filters so that real-time operation is just possible on a DVT, compared with the less-than-40-percent running time of LPC using standard digital-filter-computation structures. Progress in charge-coupled (CCD) or charge-transfer devices (CTD), however, offers the possibility of efficient channel-vocoder-filter implementations. A small study was undertaken jointly with the Electronics Research Laboratory of the University of California at Berkeley to (a) study the overall vocoder configuration, (b) design, fabricate, and test a prototype full-wave rectifier-desampling filter, (c) breadboard a discrete transversal bandpass vocoder filter, and (d) create a fully integrable operational-amplifier design compatible with the on-chip transversal-filter environment.

The overall vocoder configuration is based on a Belgard structure using finite impulse response (FIR) transversal filters. With this approach, the CTDs can implement most of the filters, with the envelope detector in the vocoder analysis one of the unknown factors. The second task of designing the full-wave rectifier-desampling filter was completed during the 3-month tasking, and the completed chip was delivered to Lincoln Laboratory for evaluation. The chip is shown in Fig. IX-1.

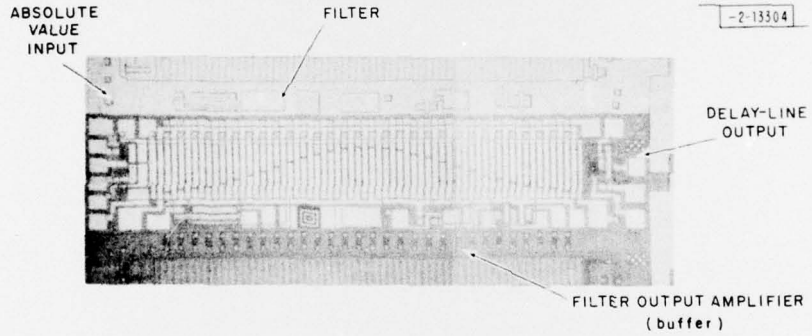


Fig. IX-1. The rectifier-desampling filter chip.

The input circuit of the chip utilizes the differential capabilities of a "fill and spill" type input[1] in order to calculate the absolute value of the applied signal. This input uses a combination of charge-coupled and bucket-brigade type transfer mechanisms to yield a structure which requires only a single level of metalization. This avoids the problem of threshold shifts which occur between different metalization levels and thus yields increased dynamic range for the absolute value circuit. In addition, the input structure was designed to automatically incorporate a bias charge level since the addition of a DC level to the input signal would degrade the accuracy of the absolute value. Figure IX-2 shows the input signal (bottom trace), the output of the absolute-value circuit (top trace), and the filter output (using absolute-value input) after 30 stages of delay through the CTD filter (middle trace).

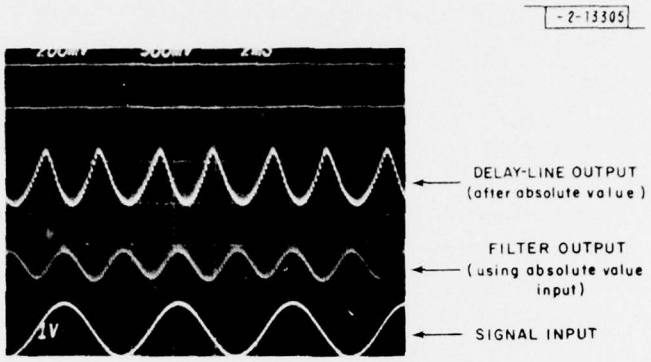


Fig. IX-2. Behavior of chip shown in Fig. IX-1.

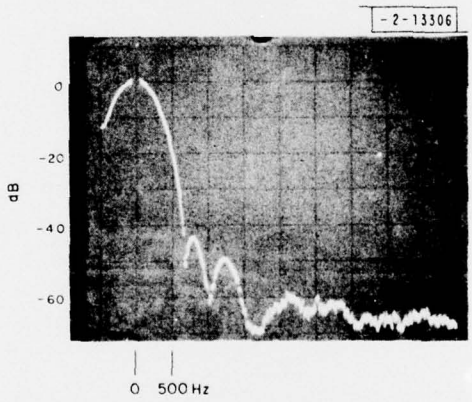


Fig. IX-3. Frequency response of filter (bypassing absolute-value input), f_c = 10 kHz.

Since the output of the absolute-value circuit is charge, it is directly compatible with the input of a CTD transversal filter. The filter was implemented using the split-electrode approach to obtain an extremely narrow low-pass-filter characteristic. The optimal linear-phase weighting coefficients[2] were determined with the constraint that the weighting coefficient be positive. All positive-weighting coefficients have two advantages: the tap-weight accuracy could be increased by a factor of two, and it made possible an extremely simple on-chip output circuit. This circuit was composed of a reset switch and a source follower, and used the sense-line capacitance to integrate the signal charge. In addition, one phase clocking was used which further reduces off-chip complexity. The filter response, shown in Fig. IX-3, has a peak side-lobe level of 45 dB down from the filter passband response. Additional measurements have shown the total harmonic distortion to be less than 1 percent. These results indicate that the complicated output circuits usually used for CTD filters[3] are not necessarily required to obtain adequate performance.

Preliminary results of task (d) concerning a fully integrable MOS operational amplifier are encouraging. The possibility of using the MOS op-amp for a sampled data-recursive filter may prove to be more efficient than the CTD transversal filter. This is a next logical area of study in the quest for an efficient channel-vocoder realization.

If the filter realization is feasible, the remaining areas of work to implement the vocoder are those involving known digital circuits technology for encoding, decoding, and pitch detection.

REFERENCES

1. C. R. Hewes, "A Self-Contained 800 State CCD Transversal Filter," Proc. CCD '75, San Diego, October 1975.
2. J. H. McClellan, T. W. Parks, and L. R. Rabiner, "A Computer *Program for Designing Optimum FIR Linear Phase Filters*," IEEE Trans. Audio Electroacoust. AU-21, 506 (1973).
3. R. D. Baertsch et al., "The Design and Operation of Practical Charge-Transfer Device Transversal Filters," IEEE Trans. Electron. Devices ED-23, 133 (1976).

X. CONTINUING WORK AND CONCLUSIONS

This report describes the FY 76-7T Lincoln Laboratory effort under the DCA Speech Evaluation contract. A separate report describes the effort undertaken on Systems Implications of Packetized Speech.

As we discussed in the Overview Section (Sec. I), the FY 76-7T effort has been directed toward "robustness" of narrowband speech terminals, improving vocoder interoperability, and implementing a low-cost LPC terminal. The work toward "robust" voice coding reported here has not yet had its impact on narrowband hardware devices, but we are within sight of that goal. Hopefully, FY 77 efforts will incorporate this year's algorithms into next year's devices. The medium-rate coding problem may yield to low-cost adaptive predictive devices, since the APC approach yields excellent quality output speech at 16 kbps. The obstacle is hardware cost and complexity. The tandem quality problem will probably remain difficult until we succeed at improving the narrow and wideband terminals independently. Our work on the low-cost LPC terminal has been very successful. We have produced two microprocessor-based LPC vocoders which can drive modems at 2.4, 3.6, and 4.8 kbps. These devices have elicited much interest from other military agencies, private agencies, and the Lincoln Laboratory Communications Division.

Our program in FY 77 continues work on narrowband speech algorithms aimed at better modeling of the speech wave, by adaptive techniques, smoothed estimates, and more accurate filter models which include zeros. A concentrated conferencing effort in FY 77 will start with an elaborate conferencing simulation facility capable of running twenty user conferences dialed up by touch-tone control. This facility will allow us to simulate all promising conferencing geometries and control strategies. Finally, we are launching a substantial effort to design and evaluate bandwidth efficient communication systems capable of voice and data transmission using the packetized virtual circuit concept.

GLOSSARY

AC	Accumulator
ADPCM	Adaptive Differential Pulse Code Modulation
ALU	Arithmetic/Logic Unit
AMDF	Absolute Magnitude Difference Function
APC	Adaptive Predictive Coding
ARC	Adaptive Residual Coding
CCD	Charge-Coupled Device
CPE	Central Processing Element
CTD	Charge-Transfer Device
CVSD	Continuously Variable Slope Delta Modulation
DCA	Defense Communications Agency
DFT	Discrete Fourier Transform
DIP	Dual In-Line Package
DRT	Diagnostic Rhyme Test
DVT	Digital Voice Terminal
ECL	Emitter Coupled Logic
EFL	Emitter Follower Logic
FDP	Fast Digital Processor
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
IC	Integrated Circuit
IFFT	Inverse Fast Fourier Transform
LDVT	Lincoln Digital Voice Terminal
LPC	Linear Predictive Coding
LPCM	Linear Predictive Coding Microprocessor
LSI	Large-Scale Integration
MMSE	Minimum Mean-Square Error
MSI	Medium-Scale Integration
PCM	Pulse Code Modulation
RAM	Random Access Memory
ROM	Read-Only Memory
SSB	Single Sideband
SSBSC	Single-Sideband Suppressed-Carrier Amplitude Modulation
SSI	Small-Scale Integration
TTL	Transistor-Transistor Logic
VLSI	Very Large-Scale Integration

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: ESD-TR-76-382	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)*: Speech Evaluation	5. TYPE OF REPORT & PERIOD COVERED: Annual Report FY 76-7T
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Bernard Gold	8. CONTRACT OR GRANT NUMBER(s): F19628-76-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Program Element No. 33126K
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Communications Agency 8th Street & So. Courthouse Road Arlington, VA 22204	12. REPORT DATE: 30 September 1976 13. NUMBER OF PAGES: ~~112~~ 105 p.
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*: Electronic Systems Division Hanscom AFB Bedford, MA 01731	15. SECURITY CLASS. *(of this report)*: Unclassified 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

speech evaluation	speech-processing systems	voice-excited systems
speech compression	Lincoln Digital Voice Terminal	pitch detection
vocoder systems	microprocessor chip sets	hybrid packaging

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This volume reports the work performed during FY 76-7T on the DCA Speech Evaluation Contract. Work during this period on System Implications of Packetized Speech for DCA is reported under separate cover. Three general areas of work are reported in this document: (1) work on narrowband terminal "robustness," (2) work on wideband-narrowband tandeming; and (3) hardware speech-terminal efforts.

The robustness issues are defined early in this report; then, work on telephone-line simulation, robust pitch extraction, and operation of LPC vocoders in acoustically noisy environments is reported.

This report also discusses some approaches and progress made in the improvement of wideband devices, and the interoperability of wideband and narrowband terminals.

The design and development of a microprocessor-based LPC vocoder, as well as some work on the development of charge-transfer-device-based channel-vocoder equipment, also are described.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

207654